Prof. Eugenia Lucia Stamate

PROVOCAREA INTERNET

Limbajul HTML si elemente de design Web

Crearea si publicarea unui site Web

General troublemaker: Alex Ricochet

http://www.softpageinternet.ro/manual_html/

Cuprins

Câteva cuvinte...înainte
HTML şi World Wide Web
Capitolul 1 Primii paşi
1. Ce este HTML?
2. Standarde şi extensii HTML
3. Conţinut şi aspect într-un document HTML
4. Uneltele de care aveţi nevoie
5. Cum să procedaţi?
6. Convenţii de notaţie
7. Rezumat

Capitolul 2 Structura unui document HTML
1. Ce este un document HTML?
2. Etichete de structură
3. Folosirea corectă a etichetelor
4. Rezumat
5. Test

Capitolul 3 Culori
1. Sistemul de definire a culorilor
2. Corespondenţa dintre codurile hexazecimal şi zecimal
3. Culoarea fundalului
4. Culoarea textului
5. Culoarea legăturilor
6. Alegerea culorilor
7. Culori "sigure"
8. Rezumat
9. Test

Capitolul 4 Fonturi
1. Formatarea caracterelor
2. Accentuarea textului
3. Etichete logice şi etichete fizice
4. Rezumat
5. Test

Capitolul 5 Formatarea textului
1. Trecerea la un rând nou
2. Titlurile (Headings)
3. Paragrafele
4. Preformatarea textului
5. Centrarea textului
6. Afişarea textului pe o singură linie
7. Blocul <DIV>
8. Linii orizontale
9. Inserarea unei adrese poştale
10. Inserarea unui citat
11. Inserarea caracterelor speciale
12. Sugestii privind aspectul textului
13. Rezumat
14. Test

Capitolul 6 Legături (Link-uri)
1. Adresa URL
2. Adrese absolute şi adrese relative
3. Stabilirea legăturilor
4. Ancore - legături în cadrul aceleiaşi pagini
5. Legătura către o pagină aflată în acelaşi director (folder)
6. Legătura către o pagină localizată în alt director (folder)
7. Legătura către pagini externe
8. Alegerea culorilor pentru legături

9. Utilizarea poştei electronice (e-mail)
10. Legături către fişiere oarecare
11. Deschiderea paginilor referite printr-o legătură
12. Crearea unei bare secundare de navigare
13. Rezumat
14. Test

Capitolul 7 Imagini şi elemente multimedia

1. Formatele fişierelor grafice
2. Câteva metode de obţinere a imaginilor
3. Inserarea unei imagini
4. Dimensionarea imaginii
5. Alinierea imaginii şi a textului
6. Imagini folosite ca fond (background) al paginii
7. Imagini folosite ca legături
8. Imaginile miniaturale (thumbnails)
9. Imaginile video
10. Sunetele
11. Sugestii privind folosirea imaginilor şi elementelor multimedia
12. Imaginile şi timpul de încărcare al paginii
13. Rezumat
14. Test

Capitolul 8 Liste

1. Liste neordonate
2. Liste ordonate
3. Imbricarea listelor
4. Liste de definiţii
5. Rezumat
6. Test

Capitolul 9 Tabele

1. Crearea unui tabel
2. Alinierea tabelului în pagină
3. Dimensionarea unui tabel
4. Spaţierea celulelor unui tabel
5. Dimensionarea celulelor unui tabel
6. Alinierea conţinutului unei celule
7. Definirea culorilor pentru un tabel
8. Titlul unui tabel
9. Capul de tabel
10. Tabele de forme oarecare
11. Celule fără conţinut
12. Grupuri de coloane
13. Atribute pentru aspectul chenarului unui tabel
14. Tabele imbricate
15. Recomandări privind folosirea tabelelor
16. Rezumat
17. Test

Capitolul 10 Cadre (Frames)

1. Documentul de definire a cadrelor
2. Cadre imbricate
3. Controlul aspectului unui cadru
4. Bare de derulare
5. Poziţionarea documentului într-un cadru
6. Cadre interne
7. Deschiderea documentelor în alte cadre
8. Sugestii privind folosirea cadrelor
9. Rezumat
10. Test

Capitolul 11 Formulare (Forms)

1. Ce este un formular?
2. Crearea unui formular
3. Eticheta <INPUT>

4. Etichetele <SELECT> şi <OPTION>
5. Eticheta <TEXTAREA>
6. Validarea datelor
7. Rezumat
8. Test

Capitolul 12 Script-uri CGI

1. Ce este CGI?
2. Transferul datelor
3. Execuţia script-ului
4. Structura unui script
5. Exemple de script-uri
6. Instalarea interpretorului Perl şi a aplicaţiei PWS
7. Rezumat

Capitolul 13 JavaScript

1. Ce este JavaScript?
2. Cum poate fi inclus un script în pagină?
3. Modul de execuţie al script-ului
4. Ce se poate realiza cu JavaScript?
5. Rezumat

Capitolul 14 Foi de stiluri (CSS)

1. Ce este un stil?
2. Tipuri de foi de stiluri
3. Efecte obţinute cu ajutorul stilurilor
4. Rezumat

Capitolul 15 Elemente avansate de grafică

1. Imagini hartă
2. Imagini animate
3. Imagini transparente
4. Rezumat

Capitolul 16 Motoare de căutare şi metatag-uri

1. Ce este un motor de căutare?
2. Metatag-uri
3. Rezumat

Crearea şi publicarea unui site

Capitolul 17 Planificarea

1. Factori care nu pot fi planificaţi
2. Etapele planificării site-ului
3. Rezumat

Capitolul 18 Designul

1. Principiile designului web
2. Organizarea unui site
3. Metodologia de construire a site-ului
4. Tehnici de design web
5. Greşeli de design
6. Rezumat

Capitolul 19 Testarea

1. Corectarea paginilor
2. Testarea paginilor
3. Rezumat

Capitolul 20 Publicarea

1. Numele de domeniu
2. Serviciul de găzduire Web (Web hosting)
3. Organizarea şi denumirea fişierelor
4. Transferul fişierelor
5. Rezumat
6. Câteva cuvinte...la final

Anexa 1 Index etichete
Anexa 2 Glosar
Anexa 3 Tabelul culorilor

Câteva cuvinte...înainte

Internetul reprezintă, fără îndoială, mediul de comunicare al viitorului, un viitor care, în multe părţi ale lumii deja a început. El reprezintă o veritabilă revoluţie paşnică şi tăcută, care rezidă în asocierea progresivă dintre informatică şi mijloacele de comunicare, conducând la o integrare fascinantă a tuturor mijloacelor tradiţionale de comunicare: telefonie clasică şi mobilă, radio, televiziune, transmisii de date, înregistrări audio şi video. Au apărut chiar forme noi de comunicare, utilizate cu entuziasm de milioane de oameni din lumea întreagă, cum sunt e-mail-ul şi chat-ul, care deja au dobândit reguli proprii, o cultură şi chiar şi un folclor propriu.

Internetul a produs totodată o extraordinară democratizare a informaţiei şi comunicării. O cantitate literalmente uriaşă de informaţie, sub toate formele sale: text, imagini statice sau video, sunet, adunate într-o "bibliotecă" accesibilă oricui, oricând, de la orice calculator legat la reţea, faţă de care chiar şi cele mai complete enciclopedii clasice par o glumă. A devenit o joacă de copil comunicarea directă, în timp real, cu verişoara din Cluj, cu prietenul din Australia, cu firma furnizoare din Brazilia sau cu banca din Elveţia. Sunt suficiente pentru toate acestea un computer, un modem şi o linie telefonică.

Deşi Internetul şi World Wide Web cunosc o expansiune exponenţială, se afirmă totuşi că se află încă în faza copilăriei. Raportându-ne la o astfel de apreciere, am putea afirma că în România Internetul este încă în faşă, dar cu un potenţial de dezvoltare cert şi rapid ascendent.

Cu toate că foarte mulţi tineri manifestă un interes şi o deschidere extraordinare către orice subiect legat de Internet, acest domeniu este cel mai adesea ignorat în licee, sau tratat cu superficialitate. Programele şcolare nu îl abordează decât tangenţial, prin urmare nu există manuale şcolare dedicate, iar lucrările de specialitate la un nivel accesibil strălucesc prin absenţă. Designul şi programarea Web se află într-o dezvoltare accelerată şi constituie o orientare profesională cu un mare potenţial de succes, dar în mod paradoxal, relativ puţini tineri se îndreaptă azi către aceste domenii. Unul din motive îl constituie, desigur, atenţia insuficientă acordată acestor ramuri ale informaticii în învăţământul liceal, alături de un contact direct cu Internetul extrem de redus, limitat mai ales de posibilităţile tehnice şi economice actuale ale şcolilor.

Argumentul principal care m-a determinat să scriu această carte a fost un sondaj realizat în rândul elevilor unui liceu de informatică, sondaj ale cărui rezultate au confirmat impresia iniţială. Şi anume faptul că, în marea lor majoritate (87%), elevii îşi doresc să înveţe nu doar să utilizeze Internetul, ci, urmând îndrăzneala tipică vârstei, să devină o prezenţă activă în acest mediu de comunicare. Cei mai mulţi îşi doresc acest lucru pentru a se exprima pe sine, pentru a comunica lumii şi în această formă, faptul că există, că au calităţi, dorinţe, aspiraţii, talente, cunoştinţe, gânduri, sentimente care merită să fie cunoscute şi de ceilalţi.

Şi cum pot fi puse toate acestea în valoare cât mai eficient, într-un mod accesibil pentru milioane de oameni din lumea largă, dacă nu pe un site Web?

Scopul acestei cărţi, dedicate limbajului specific Internetului, HTML, şi tehnicilor uzuale de creare a unui site Web, este să îi înveţe, nu doar pe tineri, ci pe toţi cei care îşi doresc acest lucru, să îşi creeze propriul site pe Internet, cu minimum de efort şi costuri, şi cu maximum de rezultate.

Cartea nu îşi propune să fie o lucrare academică, un manual exhaustiv de HTML şi Web design. Subiectele sunt extrem de vaste şi ar fi fost imposibilă cuprinderea lor într-o singură lucrare. Am dorit să ofer o imagine de ansamblu asupra limbajului HTML, cu un

marcat accent pe aspectele practice ale utilizării sale, precum şi o trecere în revistă a unora dintre cele mai folosite tehnici de programare care extind posibilităţile limbajului: script-urile CGI, JavaScript, programarea cu ajutorul foilor de stiluri CSS. De asemenea, în partea a doua a cărţii am prezentat etapele procesului de construire a unui site Web şi câteva dintre tehnicile uzuale de design Web.

Adresându-se mai ales începătorilor care stăpânesc totuşi elementele de bază (utilizarea computerului şi sistemul de operare Windows, noţiuni de editare de text şi imagini), cartea este scrisă într-un limbaj care se doreşte accesibil. Fiecare noţiune prezentată este însoţită de exemple reprezentate de secvenţe de cod HTML ce ilustrează modul ei de aplicare.
Recomandarea noastră este ca cititorii să verifice personal toate aceste exemple, deoarece numai experimentând pe cont propriu se poate ajunge la o bună înţelegere şi stăpânire a efectelor codului HTML asupra aspectului paginii Web.

Încheiem cu speranţa că aceasta carte va constitui un instrument de lucru util şi că, la finalul ei cititorii vor dobândi abilitatea de a construi, publica şi menţine un site propriu interesant, atrăgător şi perfect funcţional. Şi, de ce nu, ca o parte din ei să facă din această activitate o veritabilă profesie.

Notă asupra terminologiei utilizate

Întrucât anumiţi termeni din limba engleză, specifici domeniului abordat în prezenta lucrare, au intrat în limbajul curent ca atare, îi vom utiliza şi noi în această formă. Am optat pentru folosirea lor cu statut de neologisme, în forma originală, uneori cu forme flexionare ad-hoc, chiar cu riscul de a fi considerate drept "barbarisme" de către lingvişti puritani. În definitiv, în acest fel au intrat în limbaj termeni deja consacraţi precum *computer*, *tastă*, *monitor*, *mouse, pixel, CDROM, modem*, *server*, *port*, ca să dăm numai câteva exemple.
Pentru unii termeni am utilizat ambele forme aflate în circulaţie, cum ar fi *link – legătură, font – caracter, tag – etichetă*.
Există şi autori care folosesc numai echivalentele româneşti ale acestor termeni. Cu toate că traducerea lor respectă cel mai adesea sensul, aceste echivalente nu au reuşit să se impună în lumea informaticii, nu de puţine ori utilizarea lor conducând, în mod paradoxal, la confuzii sau nedumeriri. (Nu vorbim aici de tentativele ridicole, cum ar fi de pilda *icoană*, sau, încă şi mai rău, *iconiţă*, utilizate de unii autori pentru *icon*, care fireşte că nu pot avea vreo şansă de succes.)

Prezentăm în continuare o listă a acestor termeni. Explicaţii asupra sensului lor se găsesc în text şi/sau în glosarul prezentat la sfârşitul cărţii.

browser	*font*	*site*
director	*host, hosting*	*tag*
download	*icon*	*template*
e-mail	*link*	*Web*
folder	*script*	

HTML şi World Wide Web

Înainte de a începe prezentarea limbajului HTML, vom face câteva precizări despre semnificaţia conceptului **World Wide Web** (într-o traducere aproximativă, pânză de păianjen, cu sensul de reţea, extinsă în lumea largă). Precizările ar putea părea inutile, dar realitatea este că deşi foarte multă lume vorbeşte despre World Wide Web, puţini ştiu exact ce reprezintă. Mai mult,

datorită extinderii sale exponenţiale şi a extraordinarei sale popularităţi, adesea noţiunea este confundată cu cea de Internet. World Wide Web este însă doar o parte a Internetului.

Internetul există de mai bine de treizeci de ani. Bazele acestuia au fost puse la sfârşitul anilor ’60, ca urmare a unei iniţiative a Departamentului de Apărare al Statelor Unite, care avea drept scop realizarea unei reţele robuste şi fiabile de comunicaţii între computere aflate la distanţă. Reţeaua avea rolul de a asigura un contact permanent între elementele sale componente, chiar dacă o parte din legături ar fi fost întrerupte ca urmare a unui atac nuclear. Ulterior, reţeaua a pătruns în domeniul civil şi a fost adoptată rapid, iniţial de lumea academică, mai apoi de companiile comerciale.
Internetul nu este altceva decât un ansamblu alcătuit dintr-un număr enorm de reţele de computere interconectate, localizate pe toată suprafaţa globului, care suportă diverse pachete software cum ar fi e-mailul, grupurile de ştiri (*newsgroup*), protocoalele de transfer al fişierelor (FTP), Gopher şi World Wide Web.

Cu toate avantajele imense oferite de posibilităţile de comunicare globală a informaţiilor ca şi de rapiditatea comunicării, Internetul a rămas, vreme de două decenii, doar apanajul agenţiilor guvernamentale şi al mediilor ştiinţifice. Motivul l-a constituit dezorganizarea sa iniţială cvasi-totală. Folosirea unor standarde de comunicaţie şi a unor pachete software foarte diverse reprezentau nişte bariere tehnologice aproape insurmontabile pentru utilizatorii obişnuiţi. În plus, informaţiile accesibile la acea vreme pe Internet erau prezentate sub forma unor simple fişiere text şi nu în forma cunoscută azi. Dezvoltarea explozivă pe care a cunoscut-o Internetul în ultimii zece ani a fost determinată de apariţia World Wide Web.

World Wide Web a luat naştere în 1989, ca urmare a eforturilor unor fizicieni ai CERN (Laboratorul European de Fizica Particulelor) de a pune la punct un sistem standardizat pentru crearea şi distribuirea documentelor electronice oriunde pe glob, sistem care să permită integrarea tuturor elementelor media: text, imagini, sunet. Aşa cum dovedeşte imensa sa popularitate, Web-ul s-a dovedit a fi mediul ideal pentru distribuirea informaţiilor într-o maniera extinsă şi accesibilă. Motivul îl constituie faptul că Web-ul asigură o interfaţă **hypermedia** pentru informaţii. Hypermedia reprezintă variatele tipuri de informaţie (text, imagini, fişiere audio şi video) care pot fi localizate oriunde pe glob, împreună cu legăturile dintre ele. Pentru a transmite şi a afişa informaţiile hypermedia, Web-ul foloseşte un protocol de transfer (un set de reguli) care este numit **HTTP** (*Hypertext Transfer Protocol,* protocol de transmitere a hypertextului).
Deci, în esenţă, World Wide Web este o colecţie imensă de documente interconectate prin intermediul Internetului, care foloseşte protocolul HTTP pentru a afişa elementele hypermedia.

Pentru a face posibilă o asemenea performanţă, s-a impus necesitatea ca fiecare computer conectat la Internet să fie identificat în mod unic, printr-un element numit adresa **IP** (*Internet Protocol*). Adresa IP este formată din patru numere, fiecare mai mic de 255, despărţite între ele prin punct (de exemplu 123.67.9.201). În vreme ce computerele lucrează cu numere, oamenii preferă numele. Din acest motiv, fiecărei asemenea adrese IP i s-a asociat şi un nume. Cum sunt sute de milioane de computere în întreaga lume, a veni cu un nume unic care să diferenţieze un computer de celelalte pare aproape imposibil. Totuşi, să ne amintim că Internetul este o reţea de reţele. El este împărţit în grupuri numite **domenii** care, la rândul lor, sunt împărţite în **subdomenii**. Datorită acestui fapt, chiar dacă un computer poartă un nume destul de comun,

prin asocierea acestui nume cu domeniul şi subdomeniile din care face parte, se obţine un nume unic de identificare al computerului respectiv.

Organizarea domeniilor pe Internet respectă un model ierarhic. Primul nivel ierarhic îl constituie diversele tipuri de organizaţii. Tot pe primul nivel se află şi domeniile corespunzătoare ţărilor lumii, de exemplu *uk*, pentru Marea Britanie sau *ro*, pentru România.
Următorul nivel ierarhic îl constituie organizaţiile, firmele, instituţiile care deţin propriile domenii şi subdomenii.

Internetul conectează două tipuri de computere: **serverele**, care furnizează documentele, şi **clienţii**, care solicită şi afişează documentele pentru a fi vizualizate de utilizator.
Pentru a accesa şi afişa documentele HTML, pe computerul client rulează aplicaţii numite **browsere**. Uneori termenii server Web şi client Web pot produce confuzii, deoarece se referă atât la computerele pe care rulează aceste aplicaţii cât şi la aplicaţiile în sine. Din acest motiv, pentru aplicaţia client vom folosi termenul de *browser*, iar pentru aplicaţia sau pachetul de aplicaţii care rulează pe server, termenul de *aplicaţie server*.

Relaţia client-server pe care se bazează funcţionarea World Wide Web este facilitată de protocolul de comunicaţie HTTP. Interacţiunea dintre browserul Web şi serverul Web începe cu o cerere din partea clientului. Aplicaţia client transmite o cerere către aplicaţia server. Această cerere poate consta în transmiterea unui anumit document sau în efectuarea unei anumite tranzacţii. Aplicaţia server fie îndeplineşte cererea clientului, caz în care obiectul solicitat este transmis browserului, fie o refuză, situaţie în care browserul afişează binecunoscutele mesaje de eroare "The page cannot be displayed" sau "File not found".

Cu alte cuvinte, procesul vizualizării unei pagini Web începe cu o cerere a browserului Web către serverul Web. Browserul transmite serverului detalii despre el însuşi şi despre fişierul pe care îl doreşte prin intermediul unui **header HTTP** de cerere (headerul este acea parte a unui pachet de date care este plasată înaintea informaţiilor efectiv transmise şi care poate conţine adresa sursei şi a destinaţiei, verificări ale erorilor şi alte câmpuri). Serverul primeşte şi analizează headerele HTTP de cerere în căutarea informaţiilor relevante, cum ar fi numele fişierului care este solicitat şi transmite înapoi fişierul cerut împreună cu headerele HTTP de răspuns. Browserul foloseşte headerele HTTP de răspuns pentru a determina cum să afişeze rezultatele obţinute.

Trebuie să subliniem că browserele nu au nevoie de conexiune la Internet ca să funcţioneze. Puteţi încărca şi afişa cu ajutorul browserului dumneavoastră documente HTML care sunt stocate pe propriul hard-disk. Aceasta vă dă, printre altele, posibilitatea de a definitiva şi testa documentele HTML înainte de a le publica pe Internet, evitând astfel prezentarea unor pagini încă nefinalizate, cu erori, sau care funcţionează defectuos.

Capitolul 1
Primii paşi

1. Ce este HTML?

HTML este o abreviere de la *Hypertext Markup Language* şi reprezintă scheletul oricărei pagini de Web. HTML nu este un limbaj de programare. Nu veţi lucra aici cu variabile, expresii, tipuri de date, structuri de control. HTML este un limbaj descriptiv, prin care sunt descrise elementele structurale ale paginii de Web: titlurile, listele, tabelele, paragrafele, legăturile cu alte pagini, precum şi aspectul pe care îl are pagina din punct de vedere grafic. În fond, HTML este modul în care îi comunicam browserului ce elemente dorim să introducem în pagina Web şi care este aspectul acestora.

2. Standarde şi extensii HTML

HTML este un **limbaj standardizat**, aceasta însemnând că orice browser poate interpreta şi afişa corect un document Web. Sintaxa de bază a limbajului HTML este definită în versiunea **HTML 3.2**. În prezent a fost lansată şi versiunea HTML 4.0 care se află în curs de standardizare. De la o versiune la alta, limbajului HTML i se adăuga noi elemente. De asemenea, fiecare tip de browser (Netscape Navigator şi Internet Explorer, în particular) foloseşte anumite elemente ne-standard, pentru a îmbunătăţi capacităţile limbajului. Etichetele noi, care nu fac parte din versiunea standard poartă numele de **extensii**. În general, browserele cele mai folosite, Netscape Navigator şi Internet Explorer recunosc etichetele HTML 3.2 şi o mare parte din cele nou introduse în HTML 4.0.

Este recomandat să evitaţi extensiile, în afara cazului când aveţi un motiv foarte serios pentru a le folosi. Pentru utilizatorii browserelor care nu recunosc aceste extensii, anumite părţi ale documentului dumneavoastră pot deveni complet ilizibile.
Bineînţeles, extensiile pot afecta în grade diferite aspectul formal al paginii dumneavoastră. De exemplu, folosirea extensiilor care ţin de aspectul liniilor orizontale incluse în pagină nu va afecta în mod grav aspectul paginii în browserele care nu suportă aceste extensii. Cele mai multe dintre acestea vor ignora extensiile şi vor afişa o line orizontală standard. Pe de altă parte, extensiile care realizează formatarea şi alinierea textului şi aspectul fonturilor pot face ca documentul să aibă un aspect foarte dezordonat în browserele care nu suportă extensii de acest tip. Sau, în cazurile cele mai grave, acea parte a documentului poate să nu fie deloc afişată.

3. Conţinut şi aspect într-un document HTML

Pentru a realiza documente Web eficiente, cu impact asupra cititorului, şi pentru a valorifica pe deplin puterea limbajului HTML, trebuie să ţineţi seama de o idee fundamentală: HTML este destinat structurării documentele şi nu doar formatării în vederea afişării lor.

HTML furnizează modalităţi variate pentru a defini aspectul documentului: specificaţii legate de fonturi (tipurile de caractere), de formatare a textului, de culori, etc. Desigur, aspectul formal este important, căci el poate veni în beneficiul sau în detrimentul accesibilităţii informaţiilor prezentate.

Totuşi, nu aspectul formal, ci conţinutul documentului primează. HTML conţine numeroase căi de structurare a conţinutului documentului fără a pune problema felului cum vor fi afişate elementele: titluri, liste, paragrafe, imagini, etc. Toate acestea sunt definite în limbajul HTML fără a impune browserului să le afişeze într-un anumit mod. Browserului i se lasă astfel posibilitatea de a afişa respectivele elemente în conformitate cu rolul pe care îl au ele în structura documentului şi nu respectând regulile impuse de creatorul documentului HTML.

Marele avantaj al abordării bazate pe conţinut faţă de cea bazată pe aspect rezidă în faptul că paginile dumneavoastră vor fi vizualizate cu tipuri diferite de browsere, fiecare având versiuni diferite.

În abordarea bazată pe **aspect**, dacă browserul nu recunoaşte indicaţiile de formatare specificate, fie nu le execută, fie le execută eronat, astfel că structurarea paginii va avea de suferit.
În abordarea bazată pe **conţinut**, indiferent cum va alege browserul să execute indicaţiile de formatare, semnificaţia elementelor din pagină va rămâne neschimbată.

De exemplu, dacă veţi afişa titlurile şi subtitlurile din pagină folosind tipuri diferite de fonturi, caractere cu stiluri variate: aldine (bold, sau îngroşate), cursive (italice, sau înclinate), subliniate, etc., un browser care nu recunoaşte aceste elemente ar putea afişa titlurile în acelaşi mod în care afişează restul textului, astfel că semnificaţia lor în pagină şi-ar pierde relevanţa.
Pe de altă parte, dacă veţi folosi etichetele speciale pentru titluri, chiar daca browserul nu le recunoaşte, are posibilitatea de a evidenţia totuşi titlurile în cea mai bună manieră pe care o are la dispoziţie, astfel încât semnificaţia lor pentru restul documentului se va păstra.

4. Uneltele de care aveţi nevoie

Instrumentele minime de care aveţi nevoie pentru a realiza documente HTML sunt, în esenţă, următoarele: un **editor de text**, cu ajutorul căruia să scrieţi documentul, un **browser**, prin intermediul căruia să vizualizaţi aspectul şi conţinutul documentului, un **editor grafic** cu care să prelucraţi imaginile pe care le veţi introduce în pagină şi o **conexiune la Internet**.

Editorul

Pentru a scrie documente HTML aveţi la dispoziţie trei posibilităţi:

1. Folosirea unui simplu **editor de text**.
Cel mai cunoscut şi accesibil este Notepad-ul din Windows.

Atenţie!

Nu este recomandat să folosiţi un procesor de text complex cum este Microsoft Word, chiar dacă acesta poate salva documentele în format HTML (cu extensia *.htm*), deoarece la salvarea documentelor Word-ul adaugă automat la textul documentului dumneavoastră o mare cantitate de cod conţinând informaţii lipsite de relevanţă care încarcă inutil documentul şi măreşte dimensiunile fişierului.

2. Folosirea unui **editor HTML**.

Acest tip de editoare dedicate oferă facilităţi în scrierea codului, realizează corectarea documentului din punct de vedere sintactic şi chiar scriu secvenţe de cod pentru anumite elemente pe care doriţi să le inseraţi în document.

3. Folosirea unui **editor WYSIWYG** (*What You See Is What You Get*, în traducere liberă "ceea ce vezi la editare este ceea ce obţii la vizualizare").

Un editor de tip WYSIWYG permite alegerea şi formatarea, cu ajutorul mouse-ului, a elementelor pe care doriţi să le includeţi în pagină: tabele, link-uri, formulare, puteţi stabili tipul fonturilor, culorile, aranjarea textului în pagina. Pe măsură ce un anumit element este inserat, editorul scrie în mod automat codul HTML corespunzător, fără ca utilizatorul să fie nevoit să intervină asupra lui. Totuşi, dacă aveţi cunoştinţe de HTML puteţi interveni la nivelul codului în orice manieră doriţi, orice schimbare pe care o faceţi în cod, reflectându-se automat în aspectul paginii.

Printre profesionişti persistă o controversă în privinţa tipurilor de editoare care ar trebui folosite pentru a edita documentele HTML. În esenţă, problema este aceasta: este mai bine să-ţi construieşti paginile folosind HTML "pur", adică scriind totul tu însuţi, sau este de preferat să foloseşti un editor de tipul WYSIWYG? Editoare deosebit de puternice de acest tip: Macromedia Dreamweaver, Microsoft Front Page, Coffe Cup ca să dăm numai câteva exemple, par a înclina definitiv balanţa în favoarea acestui tip de editare. Ele oferă facilitaţi avansate de editare a paginii, inclusiv posibilitatea de a crea pagini dinamice şi de a include script-uri JavaScript sau applet-uri Java. Cu ajutorul lor se pot construi site-uri extrem de complexe.

Există, de asemenea şi editoare de tip "built-in". Acestea sunt cel mai uşor de folosit şi adesea sunt puse la dispoziţie de providerii care oferă şi servicii gratuite de hosting (www.home.ro, www.rhp.ro, etc.).
Editoarele de acest tip oferă un număr (de obicei redus) de template-uri (pagini preformatate), şi doar câteva opţiuni de editare: puteţi adăuga text, modifica tipul fonturilor, culoarea fundalului şi a textului, forma butoanelor de navigare, puteţi include un număr prestabilit de imagini. Din păcate, gradul de control asupra aspectului paginii este foarte redus. Utilizatorul este limitat la câteva opţiuni, iar numărul mic de template-uri puse la dispoziţie face ca pagina să fie lipsita de originalitate.

Aţi putea, atunci, să vă întrebaţi de ce mai este nevoie să învăţaţi HTML dacă există editoare profesionale care scriu codul în locul dumneavoastră. În fond, mulţi dintre web designerii profesionişti le folosesc, deoarece a scrie codul unui site "by hand", cu ajutorul unui simplu editor de text, este o munca titanică şi mare consumatoare de timp.

Răspunsul este simplu: trebuie să cunoaşteţi HTML deoarece oricât de performant ar fi un editor WYSIWYG, mai devreme sau mai târziu vă veţi găsi în situaţia de a interveni la nivelul codului. Cea mai comună situaţie este aceea în care, datorită multiplelor operaţii de editare cum sunt mutarea, copierea, ştergerea elementelor, anumite părţi ale paginii nu vor mai funcţiona corect.
De obicei, intervenţiile necesare asupra codului pentru depanarea sa sunt minime şi uşor de realizat... cu condiţia să ştiţi ce trebuie să faceţi. Cu alte cuvinte, să ştiţi HTML.

În concluzie, oricare ar fi opţiunea dumneavoastră, fie că veţi folosi un editor de HTML de tip text, fie că veţi folosi unul de tip WYSIWYG, pentru a putea construi documente Web corecte şi funcţionale este nevoie să stăpâniţi bine limbajul HTML.

Browserul

Evident, pentru a vizualiza documentele dumneavoastră aveţi nevoie de un browser. Recomandarea noastră este să deţineţi cel puţin două dintre cele mai răspândite browsere, Internet Explorer şi Netscape Navigator, de dorit ar fi chiar cu versiuni

diferite ale aceluiaşi browser. Testarea paginilor dumneavoastră sub diferite browsere este o etapă importantă înaintea publicării lor pe Web.

Deşi în anii de început ai Web-ului problema aspectului paginilor în diferite tipuri de browsere era vitală, deoarece erau folosite o mare diversitate de browsere, fiecare tip realizând afişarea paginilor în maniera sa proprie, în prezent această problemă şi-a pierdut din importanţă. Motivul este acela că browserele Internet Explorer au acaparat cea mai mare parte din piaţă, statisticile arătând că Internet Explorer este tipul folosit de 93% dintre utilizatori iar Netscape Navigator de circa 6% dintre ei.

Editorul grafic

Desigur, veţi dori să introduceţi imagini în paginile dumneavoastră. Pentru a putea să prelucraţi imaginile în conformitate cu intenţiile dumneavoastră şi cu rolul lor în pagină aveţi nevoie de un editor grafic.
Numărul editoarelor de acest tip este foarte mare, printre cele mai puternice şi mai cunoscute fiind CorelDraw şi Adobe Photoshop. Puteţi alege însă acel editor cu care sunteţi deja familiarizat, chiar dacă mai puţin performant, mai ales dacă pentru început nu doriţi să faceţi prelucrări foarte complexe ale imaginilor.

Conexiunea la Internet

Spre deosebire de primele trei instrumente, conexiunea la Internet nu este absolut esenţială în etapa de construire a documentului HTML. Aşa cum spuneam, documentele HTML se pot construi şi vizualiza local, pe propriul computer. Cu toate acestea, existenţa unei conexiuni Internet vă va ajuta foarte mult în procesul de învăţare, şi nu numai.
Pe de-o parte deoarece pe Web veţi găsi nenumărate exemple, şi bune şi rele, pe care le puteţi studia. În plus, veţi avea acces la o cantitate uriaşă şi în continuă creştere de resurse, documentaţie şi produse care vă vor ajuta cu siguranţă la construirea paginilor dumneavoastră. Şi, în fine, accesul la Internet este esenţial în etapa de testare a legăturilor cu pagini externe.

5. Cum să procedaţi?

Pentru a folosi eficient informaţiile din această carte iată mai jos câteva indicaţii în privinţa modului în care vă puteţi organiza activitatea:

- Creaţi-vă în folderul *My Documents* (sau într-o alta locaţie aleasă de dumneavoastră) un folder de lucru numit *Work*.

- Deschideţi editorul de text în care aţi ales să lucraţi (ar fi de dorit ca pe parcursul procesului de învăţare să folosiţi un simplu editor text, de exemplu Notepad).

- Scrieţi (editaţi) fiecare exemplu pe care îl găsiţi aici.

- Salvaţi-l în folderul dumneavoastră cu numele indicat în blocul `<TITLE>` şi extensia *.html*. Puteţi folosi atât extensia *.html* cât şi extensia *.htm* dar este bine ca odată ce aţi ales o anumită extensie să fiţi consecvent în folosirea ei. Denumiţi-vă fişierele cu litere mici.

Atenţie!

Editorul de text *Notepad* salvează fişierele cu extensia implicita "*.txt*". De aceea când salvaţi este necesar să precizaţi în mod explicit extensia dorită (*.html* sau *.htm*)

- Când veţi salva fişierul creat în Notepad s-ar putea să constataţi că restul de fişiere HTML pe care le-aţi salvat anterior nu apar în fereastră. Pentru a le

vedea, deschideţi meniul *File>Save As...*, iar în fereastra deschisă selectaţi din listă *All files*.

- Deschideţi cu browserul utilizat fişierul HTML (*File> Open> Browse*) pentru a vedea cum arată pagina Web descrisă în fişierul dumneavoastră.

- După fiecare modificare pe care o faceţi în fişierul HTML nu uitaţi să-l salvaţi, altminteri schimbările făcute nu se vor reflecta în aspectul final al paginii.

- Pentru a observa felul cum modificările pe care le faceţi în codul HTML schimbă aspectul paginii, trebuie să apăsaţi butonul *Refresh/Reload* al browserului.

- Pentru a putea să vizualizaţi cu uşurinţă aspectul paginilor dumneavoastră este bine să aveţi deschise permanent atât editorul de text în care scrieţi documentul HTML cât şi browserul cu care îl vizualizaţi.

6. Convenţii de notaţie

HTML nu este un limbaj *case-sensitive*, adică nu face distincţia între literele mici şi cele mari(majuscule). Aceasta înseamnă că indiferent cum scriem etichetele, cu litere mari sau cu litere mici, ele vor fi corect interpretate de browser. Din acest motiv pe parcursul acestei lucrări vom utiliza o convenţie de notaţie în care etichetele (*tag*-urile) HTML sunt scrise cu majuscule iar atributele asociate acestora sunt scrise cu litere mici pentru o mai bună diferenţiere şi evidenţiere a acestora. Desigur că în documentele pe care le scrieţi puteţi respecta această convenţie sau puteţi scrie cu un singur tip de litere, după cum vă este mai uşor.
Peste tot în cadrul textului, unde apar etichete, atribute sau valori ale acestora am folosit caractere Courier New pentru a diferenţia elementele de limbaj de restul textului.

De asemenea, veţi observa că, în exemplele prezentate, codul HTML este scris **indentat**, adică aliniat la diverse nivele. Ca şi în cazul etichetelor, acest lucru nu are importanţă pentru browser. Indentarea vă este utilă doar dumneavoastră, pentru a diferenţia şi urmări mai uşor diversele elemente pe care le includeţi în document. Totuşi, dacă în cazul etichetelor puteţi opta pentru orice variantă doriţi în privinţa scrierii lor, indentarea este recomandată chiar şi programatorilor cu experienţă. Un cod neindentat este foarte greu de citit şi de depanat, în cazul apariţiei unor erori.

7. Rezumat

World Wide Web este o colecţie imensă de documente interconectate prin intermediul Internetului, care asigură o interfaţă hypermedia pentru informaţii.

Pentru a transmite şi afişa documentele, World Wide Web foloseşte protocolului de comunicaţie HTTP. Funcţionarea World Wide Web se bazează pe relaţia client-server. Clienţii sunt browserele Web iar serverele sunt serverele Web.

Limbajul HTML realizează descrierea documentelor Web. Prin intermediul său i se comunică browserului ce elemente fac parte din pagina Web şi care este aspectul acestora. Este recomandată crearea unor documente HTML orientate către conţinut şi nu către aspect.

Pentru a crea documente HTML aveţi nevoie de un set minimal de instrumente: un editor de text, un browser, un editor grafic şi o conexiune la Internet.

Capitolul 2
Structura unui document HTML

1. Ce este un document HTML?

Un **document HTML** nu este altceva decât un fişier text care, pe lângă textul propriu-zis ce va apărea în pagină, conţine şi o serie de elemente speciale, denumite **etichete**, sau marcaje (*tags*, în limba engleză).

Notă

Toate celelalte elemente, de tip multimedia, care însoţesc documentul HTML, cum sunt sunetul, imaginile video, grafica, etc., nu fac parte din structura acestuia. Documentul HTML include referinţe la aceste elemente, prin intermediul unor etichete speciale, care indică browserului modul în care ele vor fi încărcate şi integrate în pagina Web.

Etichetele nu sunt instrucţiuni în sensul cunoscut al noţiunii, ele având doar rolul de a comunica browserului semnificaţia şi modul de afişare al elementelor incluse, privind aspectul textului, al fonturilor (tipurilor de caractere) şi în general, al tuturor elementelor prezente în pagină. Etichetele sunt delimitate de perechi de paranteze unghiulare `"< >"`.

Unele etichete permit precizarea anumitor caracteristici ale elementului pe care îl introduc în document prin intermediul unor perechi *caracteristică - valoare* numite **atribute**. Există atribute specifice doar anumitor elemente şi atribute ce sunt utilizate în asociere cu mai multe etichete. O etichetă poate avea unul, nici unul, sau mai multe atribute. La rândul lor, atributelor li se pot atribui **valori** diferite.

Atât etichetele cât şi atributele sunt *case-insensitive*, ca atare nu va exista nici o diferenţă de acţiune între, să spunem, `<head>`, `<Head>`, `<HEAD>`, sau chiar `<HeaD>`. Ele sunt echivalente. Pe de altă parte, valoarea unui atribut poate fi *case-sensitive*, cum, în principiu, este cazul locaţiilor fişierelor şi adreselor URL.

Regulile de folosire a etichetelor, atributelor şi valorilor acestora reprezintă **sintaxa** limbajului HTML. Similar limbajelor de programare, în HTML respectarea sintaxei limbajului este determinantă pentru o bună funcţionare a documentului.
Spre deosebire de limbajele compilate cum ar fi limbajul C, în care programele trec înainte de execuţie prin faza de compilare, etapă în care compilatorul semnalează eventualele greşeli de sintaxă, la documentele HTML nu se întâmplă acest lucru. Documentele HTML sunt interpretate de browser exact aşa cum au fost ele scrise. Prin urmare, orice greşeală de sintaxă se va reflecta direct în aspectul paginii Web, conducând, de cele mai multe ori, la o funcţionare defectuoasă a acesteia.
Dacă la vizualizarea paginii dumneavoastră cu browserul constataţi că anumite secvenţe ale paginii nu au aspectul aşteptat, sau elementele pe care doriţi să le introduceţi nu sunt afişate, reveniţi asupra documentului HTML şi verificaţi încă o dată sintaxa acestor elemente.

HTML lucrează într-o manieră foarte uşor de înţeles. În esenţă trebuie să scrieţi textul şi să precizaţi elementele care doriţi să apară în pagină şi să le includeţi între anumite etichete specifice. De exemplu, dacă dorim să afişăm o propoziţie

cu caractere îngroşate (bold), vom marca începutul acesteia folosind eticheta <B> iar pentru marcarea sfârşitului propoziţiei eticheta </B>.

```
<B> Acesta este un text scris cu litere aldine</B>
```

Atenţie!

Browserul nu ţine cont de numărul spaţiilor dintre cuvinte, afişarea făcându-se invariabil cu un singur spaţiu. De asemenea, nu se poate preciza mărimea liniilor de text, acestea fiind de mărimea ferestrei browserului (prin redimensionarea ferestrei se vor rearanja şi liniile de text).

După cum vom vedea mai târziu, se poate preciza totuşi unde să se termine un anumit rând şi cum se poate alinia un text în pagină.

Etichetele HTML sunt de două tipuri:

- etichete **container** (*container tags*)
- etichete **vide** (*empty tags*)

Etichetele **container** sunt de forma:

```
<TAG> bloc de text </TAG>
```

unde:
<TAG> - marchează începutul unui bloc
</TAG> - marchează sfârşitul blocului

Etichetele specifică formatul în care va fi afişat textul conţinut între eticheta de început şi cea de final. Majoritatea etichetelor sunt de acest tip. O particularitate a etichetelor container este că, în cazul unora dintre ele, prezenţa etichetei de închidere (</TAG>) este opţională. Pe parcursul capitolelor care vor urma, vom preciza explicit care anume sunt etichetele care au această proprietate.

Etichetele **vide** au forma:

```
<TAG>
```

Acest tip de etichete nu se referă la formatul textelor, ci la introducerea anumitor elemente, ca de exemplu: paragrafe, sfârşit de linie, linii orizontale şi altele, deci dau indicaţii browserului despre ce element este vorba şi despre cum să afişeze acel element.
Etichetele vide nu au etichetă de închidere.

2. Etichete de structură

Orice document HTML conţine două secţiuni:
- antetul (*head*)
- corpul documentului (*body*)

Structura generală a unui document HTML este următoarea:

```
<HTML>
  <HEAD>
    <TITLE> </TITLE>
  </HEAD>
  <BODY>
```

```
  </BODY>
</HTML>
```

Documentul este delimitat de etichetele <HTML> </HTML> şi este format din cele două părţi:

- **antet** (*head*) - este delimitat de etichetele <HEAD> </HEAD> şi conţine titlul documentului precum şi alte informaţii privind documentul. În antetul documentului se mai inserează şi anumite secvenţe de program (script-uri), care se execută la încărcarea documentului în browser.

- **corp** (**body**) - care delimitează conţinutul propriu-zis al documentului şi este inclus între etichetele <BODY> </BODY>

Iată acum şi semnificaţia etichetelor menţionate şi care fac parte din structura oricărui document HTML:

<HTML> </HTML>
Între aceste etichete este inclus întregul document HTML. Ele comunică browserului unde începe şi unde se termină documentul.

<HEAD> </HEAD>
Între aceste etichete sunt incluse titlul, deja menţionat, precum şi alte informaţii despre documentul HTML. Aceste elemente, numite *metatag*-uri sunt deosebit de importante pentru ca pagina să fie cât mai bine cotată de către motoarele de căutare. Ele vor face obiectul unui capitol separat.
Metatag-urile şi antetul în ansamblul său nu sunt vizibile pentru vizitator în pagina Web. Totuşi, la fel ca întreg codul HTML al paginii, antetul poate fi vizualizat selectând din meniul browserului opţiunea *View > Source.*

<TITLE> </TITLE>
Stabileşte **titlul** documentului HTML.
Titlul documentului este deosebit de important deoarece este unul din criteriile folosite de motoarele de căutare pentru indexarea paginii. Prin urmare, străduiţi-vă să găsiţi un titlu care să descrie cât mai corect şi complet conţinutul paginii dumneavoastră.

Atenţie!

Eticheta de mai sus nu stabileşte titlul care apare în cadrul paginii, ci pe acela care apare obişnuit în bara de titlu a browserului.

<BODY> </BODY>
Conţine descrierea textului şi elementelor paginii. În corpul documentului se stabilesc, deci, aspectul şi conţinutul paginii Web. Elementele conţinute în această secţiune sunt vizibile în pagină.

Poate nu vă vine să credeţi, dar acum aveţi deja posibilitatea de a crea o pagină foarte simplă de text. Aveţi Notepad-ul deschis? Dacă nu, deschideţi-l acum şi scrieţi exemplul următor:

Exemplul 2. 1

```
<HTML>
  <HEAD>
    <TITLE>Pagina mea</TITLE>
  </HEAD>
  <BODY>
    Am reusit sa scriu prima mea pagina de Web. Urmatoarea va fi cu mult
    mai buna.
```

```
  </BODY>
</HTML>
```

Salvaţi fişierul cu numele *primapagina.html*. Apoi deschideţi browserul pe care îl folosiţi (Internet Explorer sau Netscape), selectaţi meniul *File > Open > Browse*, căutaţi folderul în care aţi salvat fişierul, deschideţi-l şi vă veţi afla în faţa primului dumneavoastră document HTML : Figura 2.1

Am reusit sa scriu prima mea pagina de Web. Urmatoarea va fi cu mult mai buna.

Dacă doriţi să scrieţi un text pe mai multe linii în pagină, va trebui să folosiţi eticheta
 (de la *line break*), care face trecerea pe o linie noua. Eticheta <BR> este de tip *empty*, deci nu are etichetă de închidere. Încercaţi exemplul de mai jos, în care afişarea textului documentului se va face pe două rânduri.

Exemplul 2. 2

```
<HTML>
  <HEAD>
    <TITLE>Pagina mea</TITLE>
  </HEAD>
  <BODY>
    Am reusit sa scriu prima mea pagina de Web.<BR>Urmatoarea va fi cu
    mult mai buna.
  </BODY>
</HTML>
```

Acum, documentul dumneavoastră va arăta ca în Figura 2.2

Am reusit sa scriu prima mea pagina de Web.
Urmatoarea va fi cu mult mai buna.

Deoarece structura documentelor HTML obişnuite este în realitate mult mai complexă, este recomandat ca, pe măsură ce procesul de editare avansează, să introduceţi comentarii în text pentru a face această structură cât mai explicită. Comentariile nu sunt vizibile pentru browser, ele servesc creatorului paginii pentru a da un grad crescut de lizibilitate documentului HTML.
Comentariile constau în scurte informaţii, explicaţii la elementele de cod utilizate, etc., fiind deosebit de utile ulterior, mai ales în situaţia în care se impune depanarea codului.
Comentariile se introduc prin includerea textului între etichetele de mai jos.

```
<!-- -->
```

Iată două exemple:

```
<!- - Acesta este un comentariu introdus in pagina Web - ->

<!—Acesta este un comentariu
  pe mai multe randuri
  care ia sfarsit aici -->
```

3. Folosirea corectă a etichetelor

Etichetele container se compun, aşa cum am văzut mai devreme, din perechi de etichete: etichetă de deschidere <TAG> şi de închidere </TAG>.
La construirea unei pagini Web complexe veţi fi pus adesea în situaţia de a folosi mai multe etichete pentru aceeaşi secvenţă de text.
De pildă, veţi dori să atribuiţi unui text două proprietăţi: litere aldine sau îngroşate (*bold*) şi text centrat în pagină. Pentru aceasta va trebui să utilizaţi două perechi de etichete, care descriu cele două proprietăţi, în mod simultan, metodă numită **imbricarea** etichetelor (*nested tags*), ca în exemplul de mai jos:

```
<CENTER><B>Text cu aldine si centrat</B></CENTER>
```

Este foarte important ca, în cazul folosirii mai multor etichete container împreună, ele să fie plasate în mod corect.

În această situaţie, principiul de utilizare este: "**Last In - First Out**" (LIFO). Acest lucru înseamnă că ultima etichetă deschisă trebuie să fie prima care se închide.

Atenţie!

Trebuie să fiţi foarte atent la închiderea etichetelor, pentru a nu încălca principiul LIFO. În caz contrar, secvenţa de cod din pagină nu va funcţiona corect.

Iată un exemplu generic de folosire corectă a două etichete:

```
<TAG1><TAG2>Etichete utilizate corect</TAG2></TAG1>
```

şi un altul de folosire incorectă:

```
<TAG1><TAG2>Etichete utilizate incorect</TAG1></TAG2>
```

4. Rezumat

Un document HTML este un fişier text care conţine textul care va apărea în pagină şi etichete. Acestea au rolul de a comunica browserului semnificaţia şi modul de afişare al elementelor incluse între ele.

Etichetele sunt însoţite de atribute care sunt perechi *caracteristică* - *valoare* şi care au rolul de a stabili diverse caracteristici ale etichetei. Etichetele pot fi de două tipuri: etichete container şi etichete vide. Etichetele container necesită eticheta de închidere în timp ce etichetele vide nu trebuie închise.

Un document HTML este format din antet şi corp. Etichetele de structură ale documentului sunt:
- <HTML> </HTML> care delimitează documentul
- <HEAD> <HEAD> care delimitează antetul
- <TITLE> </TITLE> care delimitează titlul documentului
- <BODY> </BODY> care delimitează corpul documentului.

Într-un document HTML se pot introduce comentarii prin introducerea textului între etichetele <!-- -->

Imbricarea etichetelor respectă regula LIFO: prima etichetă deschisă este ultima care se închide.

5. Test

1. *HTML este un acronim de la:*

a) Hyper Text Marker Line
b) Hyper Text Markup Language
c) Hyper Technical Method Library

2. *Etichetele HTML sunt incluse între:*

a) [şi]
b) (şi)
c) < şi >

3. *Etichetele şi textul care nu sunt vizibile în pagină sunt plasate în blocul:*

a) `Body`
b) `Head`
c) `Table`

4. *Ce fel de program este necesar pentru a scrie HTML?*

a) un editor de text
b) un editor grafic
c) HTML Development 4.0

5. *O pagina Web este formată din două părţi:*

a) `Top` şi `Bottom`
b) `Body` şi `Frameset`
c) `Head` şi `Body`

6. *Care etichete îi comunica browserului unde începe şi unde se termină pagina?*

a) `<HTML>`
b) `<HEAD>`
c) `<BODY>`

7. *Care dintre următoarele elemente nu poate fi găsit în secţiunea Head?*

a) `Title`
b) `Table`
c) `Metatag`

8. *În construcţia următoare:*

```
<TITLE>Pagina mea</TITLE>
```

Pagina mea *reprezintă:*

a) Numele fişierului HTML
b) Titlul care va apărea în pagina Web
c) Titlul care va apărea în bara de titlu a ferestrei browserului

9. *Care dintre următoarele fişiere nu este un fişier HTML?*

a) mypage.htm
b) mypage.txt
c) mypage.html

10. *Iată exemplul de mai jos:*

```
<TAG1>
  <TAG2>Text</TAG2>
  <TAG3>Text
    <TAG4>Text</TAG4>
  </TAG3>Text
</TAG1>
```

Este un exemplu de folosire corectă a etichetelor?

a) Nu, deoarece închiderea etichetelor nu respectă regula LIFO.
b) Nu, deoarece nu avem voie să folosim mai mult de două etichete imbricate
c) Da, deoarece etichetele se închid corect.

Raspunsuri
1. b)
2. c)
3. b)
4. a)
5. c)
6. a)
7. b)
8. c)
9. b)
10. c)

Capitolul 3
Culori

Culoarea reprezintă un caracteristică extrem de importantă a unei pagini Web. Utilizată cu grijă şi măsură, ea poate îmbunătăţi substanţial aspectul paginii, făcând-o mai lizibilă şi mai atractivă, iar textul mai uşor de parcurs.
Pe de altă parte, utilizarea excesivă sau inadecvată a culorilor poate transforma un text de calitate într-un fel de caleidoscop obositor, greu de urmărit, şi care îi va determina pe mulţi vizitatori să renunţe la a-l mai parcurge.
Datorită faptului că ea reprezintă un atribut al multor elemente ale documentelor Web (text, fundaluri, margini, link-uri), am optat pentru prezentarea detaliată a modului de utilizare al culorilor încă din această etapă a lucrării noastre.

1. Sistemul de definire a culorilor

Sistemul general utilizat pentru definirea culorilor este **RGB** prin care sunt specificate pentru fiecare culoare care sunt cantităţile de roşu (*Red*), verde (*Green*) şi albastru (*Blue*) care o compun.

În HTML orice culoare este desemnată printr-un cod de 6 cifre hexazecimale, dintre care primele două reprezintă cantitatea de roşu, cele două din mijloc, cantitatea de verde, iar ultimele două cantitatea de albastru.

Sistemul hexazecimal este un sistem de numeraţie care foloseşte 16 cifre, existând următoarea corespondenţă între cifrele hexazecimale şi cele zecimale:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

De exemplu, #FF0000 este codul pentru roşu. Semnificaţia acestui cod este următoarea: cantitatea de roşu este maximă, iar cantităţile de verde şi albastru sunt egale cu 0. Codul pentru verde pur este #00FF00 iar pentru albastru este #0000FF.

Deoarece uneori definirea culorii prin intermediul codului său hexazecimal este incomodă, standardul HTML 3.2 stabileşte un set de 16 culori standard.
În acest caz, pentru valoarea atributului de culoare se utilizează direct numele culorii, ca în exemplul de mai jos, în care se setează culoarea magenta pentru linkurile vizitate:

```
<BODY vlink="magenta">
```

Numele şi codul culorilor stabilite în standardul HTML3.2 sunt următoarele:

Tabelul 3. 1

Nume culoare	**Cod hexazecimal**
aqua	#00FFFF
black	#000000
blue	#0000FF
fuchsia	#FF00FF
gray	#808080
green	#008000
lime	#00FF00
maroon	#800000
navy	#000080
olive	#808000
purple	#800080
red	#FF0000
silver	#C0C0C0
teal	#008080
yellow	#FFFF00
white	#FFFFFF

Mai trebuie menţionat faptul că se pot utiliza, pe lângă culori, 100 de nuanţe de gri (*gray*). Ele sunt numerotate de la 1 la 100, "gray 1" fiind nuanţa cea mai închisă, iar "gray 100" cea mai deschisă. În Anexa 3 veţi găsi tabelul culorilor însoţite de codurile lor hexazecimale.

2. Corespondenţa dintre codurile hexazecimal şi zecimal

Multe dintre editoarele grafice folosesc pentru a defini culorile codul zecimal. Deşi se bazează şi ele pe acelaşi sistem RGB de definire a culorilor, numerele care stabilesc cantităţile celor trei culori sunt specificate în sistemul zecimal.

De exemplu, o culoare care are codul hexazecimal #63FF80 va avea codul zecimal 99, 255, 128. Această corespondenţă se realizează simplu, trecând fiecare dintre cele trei numere hexazecimale din codul culorii în sistemul zecimal:

$63_{(16)}=99_{(10)}$, $FF_{(16)=}255_{(10)}$, $80_{(16)}=128_{(10)}$

Motivul pentru care trebuie să cunoaşteţi această corespondenţă este acela că pentru a folosi în documentul HTML o culoare al cărei cod este dat în sistemul zecimal, aceasta trebuie convertită în cod hexazecimal.

Conversia se poate realiza foarte simplu folosind calculatorul din Windows (*Start > Accessories > Calculator*) care trebuie setat pe opţiunea *Scientific* din meniul *View*. Veţi observa o serie de butoane radio dintre care ne interesează două: *Hex* şi *Dec*. Nu aveţi altceva de făcut decât să selectaţi *Dec*, să tastaţi numărul în baza 10 şi apoi să selectaţi opţiunea *Hex*. Rezultatul conversiei va apărea pe ecran. Conversia inversă se realizează în mod similar.

Şi acum să trecem la lucruri mai atractive şi să dăm viaţă paginii noastre adăugându-i culori. Vom lua pe rând atributele etichetei `<BODY>`. Cu ajutorul lor se pot seta culorile fundalului, textului şi linkurilor:

`bgcolor` - culoarea fundalului
`text` - culoarea textului
`link` - culoarea linkurilor
`vlink` - culoarea linkurilor vizitate
`alink` - culoarea linkului activ

3. Culoarea fundalului

Pentru a stabili culoarea fundalului unei pagini folosim atributul **`bgcolor`** (*background color*) al etichetei `<BODY>`, căruia îi atribuim o valoare astfel:

```
<BODY bgcolor="#RGB sau nume_culoare">
```

Se poate folosi drept valoare pentru atributul `bgcolor` fie codul hexazecimal al culorii fie numele ei. Iată un exemplu de pagină cu fundal verde:

Exemplul 3. 1

```
<HTML>
  <HEAD>
    <TITLE>culori1</TITLE>
  </HEAD>
  <BODY bgcolor="#00FF00">
    <H1 align="center">Pagina cu fundal verde</H1><HR>
  </BODY>
</HTML>
```

Aspectul paginii va fi cel din Figura 3.1.

4. Culoarea textului

Pentru a seta culoarea textului din întreaga pagină se foloseşte atributul **`text`** al etichetei `<BODY>`, conform sintaxei:

```
<BODY text="#RGB sau nume_culoare">
```

Dacă am stabilit o culoare pentru textul din pagină şi dorim să utilizăm o altă culoare pentru o anumită secţiune a textului (culoarea unui bloc de text), vom folosi o etichetă despre care vom vorbi pe larg în capitolul următor: eticheta `<FONT>`. Aceasta este o etichetă container care cuprinde între etichetele de început şi de final textul a cărui culoare dorim să o modificăm. Stabilirea culorii se face folosind atributul `color` al etichetei conform sintaxei:

```
<FONT color="#RGB sau nume_culoare">Text</FONT>
```

În Exemplul 3.2 culoarea textului este albastru iar anumite cuvinte sunt colorate în roşu. Aspectul paginii va fi cel din Figura 3.2.

Text albastru şi roşu

Textul din aceasta pagina este albastru cu mici exceptii

Exemplul 3. 2

```
<HTML>
  <HEAD>
    <TITLE>culori2</TITLE>
  </HEAD>
  <BODY bgcolor="#FFFFFF" text="#0000FF">
    <H1 align="center">Text albastru si rosu</H1>
    <HR>
    Textul din aceasta pagina este albastru cu mici
    <FONT color="#FF0000"> exceptii</FONT>
  </BODY>
</HTML>
```

5. Culoarea legăturilor

În general legăturile (*links*) dintr-o pagină Web au culori prestabilite (*default*), astfel:

- `blue` (albastru) - pentru legături
- `red` (roşu) - pentru legătura activa (cea pe care este fixat cursorul mouse-ului, cu butonul stâng apăsat)
- `purple` (violet) - pentru legăturile vizitate, cele pe care s-a efectuat cel puţin un *click*

Pentru a schimba culorile prestabilite se folosesc următoarele atribute ale etichetei `<BODY>`:

link pentru legături
alink pentru legăturile active
vlink pentru legăturile vizitate

conform sintaxei:

```
<BODY link="#RGB sau nume_culoare"
alink="#RGB sau nume_culoare" vlink="#RGB sau nume_culoare">
```

Documentul din Exemplul 3.3 creează o pagină în care textul este negru iar legăturile au culoarea verde, legăturile active culoarea albastru şi cele vizitate, culoarea roşu. În exemplu este folosită eticheta <A> şi atributul sau **href**, care vor fi prezentate pe larg în capitolul despre legături. În acest moment singurul lucru care ne interesează este cum putem stabili culorile acestora.

Exemplul 3. 3

```
<HTML>
  <HEAD>
    <TITLE>culori3</TITLE>
  </HEAD>
    <BODY bgcolor="#FFFFFF" text="#000000" link="#00FF00" alink="#0000FF"
    vlink="#FF0000">
      <H1 align="center">Legaturi colorate</H1>
      <HR>
      <A href="culori1.html">Legatura catre primul exemplu</A>
    </BODY>
</HTML>
```

Aspectul paginii va fi cel din Figura 3.3.

6. Alegerea culorilor

Sunt multe elemente care trebuie luate în considerare la alegerea culorilor pentru o pagină Web. Primul şi cel mai important este acela de a face conţinutul uşor de citit. Aceasta înseamnă...aţi ghicit! Litere negre pe fond alb.

Deşi în vestimentaţie combinaţia de roşu cu bleumarin este rafinată, într-o pagină Web literele roşii pe fond albastru nu arată bine şi sunt greu de citit. O altă combinaţie nefericită este cea cu fondul negru sau de culoare foarte închisă şi textul de culoare albă.

Să nu uităm nici faptul că toate monitoarele dispun de reglaje ale strălucirii luminoase şi ale contrastului pe care utilizatorii le setează diferit, în funcţie de preferinţele personale, de toleranţa ochiului, de iluminarea camerei, etc.
Pe un monitor cu reglaje medii ale contrastului şi strălucirii, un text de culoare roşie dispus pe un fundal negru va fi ilizibil, chiar dacă la valori maxime ale acestor caracteristici combinaţia poate genera efecte interesante.

În alegerea culorilor pentru un site este bine să ţineţi seama de câteva elemente de psihologia culorilor, pentru a sublinia astfel mesajul pe care vreţi să îl transmiteţi.

Iată câteva dintre sentimentele pe care le sugerează culorile:

- Roşu - agresivitate, pasiune, putere, vitalitate

- Roz - feminitate, inocenţă, moliciune
- Portocaliu - amuzament, veselie, căldură, exuberanţă
- Galben - sentimente pozitive şi cordialitate
- Verde - linişte, sănătate, prospeţime
- Albastru - autoritate, demnitate, securitate, încredere
- Violet - sofisticare, spiritualitate, mister
- Maro - utilitate, legătura cu pământul, bogăţie
- Alb - puritate, încredere, modernitate, rafinament
- Gri - sobrietate, autoritate, simţ practic
- Negru - seriozitate, distincţie, hotărâre

Alegerea culorilor pentru site trebuie să fie în concordanţă cu tema site-ului şi să sublinieze mesajul său. Astfel, dacă doriţi să construiţi un site destinat copiilor veţi folosi culori vesele şi luminoase, în vreme ce pentru un site destinat prezentării unei firme de consultanţă financiară sunt potrivite culori sobre şi care sugerează seriozitate şi încredere.

Alegerea culorilor pentru legături este, de asemenea, importantă. Legăturile au, în mod prestabilit, anumite culori. Schimbarea acestora, deşi posibilă, îl poate induce în eroare pe vizitator. Ca şi în cazul textului, alegeţi pentru legături, legăturile vizitate şi legătura activă culori care să vină în contrast cu culoarea sau imaginea de fundal a paginii. Rămâneţi consecvent culorilor alese pentru legături, pe parcursul întregului site.

7. Culori "sigure"

Aşa cum era de aşteptat, nu toate monitoarele pot reproduce cele 16.777.000 de culori furnizate de codul RGB. Cele realizate cu tehnologii mai vechi, sau cele moderne setate pentru numai 256 de culori pot reproduce corect doar un set limitat de 256 de culori, aşa-numită paletă Web sau "culorile sigure Web".

Culorile "sigure" sunt cele definite prin standardul HTML 3.2 prezentat în tabelul de mai sus sau sunt realizate prin combinaţii ale următoarelor numere hexazecimale: 00 33 66 99 CC FF
Codul hexazecimal variază între #FFFFFF (alb) şi #000000 (negru).

Un monitor setat să afişeze 256 de culori va încerca afişarea unei culori din afara acestui set prin fenomenul numit "*dithering*" (alternarea unui pixel de o culoare cu un pixel de altă culoare), ceea ce uneori dă rezultate, dar adesea rezultatul e departe de culoarea reală. Alteori, computerul va afişa culoarea "sigură" cea mai apropiată de cea originală.

În principiu, o astfel de problemă poate părea minoră. În realitate, presupunând că o pagină Web conţine atât textul cât şi fundalul ambele în culori "nesigure", prin modificarea lor de către computerul cititorului, nuanţele "aproximate" pot face textul total ilizibil.

Pentru a evita astfel de surprize neplăcute este suficient să vă menţine în limita celor 256 de culori "sigure".
Desigur, puteţi folosi o combinaţie hexazecimală ciudată, ca aceasta: #12EC8B, dar nu veţi avea nici o garanţie în privinţa modului în care va fi afişată această culoare în diferite browsere şi pe diferite platforme.

8. Rezumat

Culorile într-un document HTML sunt definite cu ajutorul codului RGB care exprimă în sistem hexazecimal cantitatea de roşu, verde şi albastru prezentă în fiecare culoare.

Pentru a stabili culorile într-o pagină Web se folosesc atributele etichetei <BODY> după următoarea sintaxă:

```
<BODY bgcolor="#RGB sau nume_culoare" text="#RGB sau nume_culoare"
link="#RGB sau nume_culoare" alink="#RGB sau nume_culoare" vlink="#RGB sau
nume_culoare">corpul documentului</BODY>
```

Alegerea culorilor pentru o pagină Web trebuie să fie în strânsă legătură cu mesajul transmis de aceasta şi să permită citirea cu uşurinţă a informaţiilor prezentate.

Pentru a evita afişarea defectuoasă a culorilor din pagină este recomandată folosirea culorilor "sigure".

9. Test

11. *Sistemul pentru definirea culorilor este:*

a) Red, Green, Blue
b) Red, Yellow, Blue
c) White, Black, Gray

12. *Culorile în HTML pot fi specificate folosind:*

a) Codul zecimal
b) Codul hexazecimal
c) Codul Morse

13. *Care este cea mai mare cifră în sistemul hexazecimal?*

a) 15
b) F
c) A

14. *O culoare care are în sistemul zecimal codul RGB cu valorile: 46, 250, 34 are codul hexazecimal:*

a) #2EFA22
b) #3CCD45
c) #66FF90

15. *Pentru a stabili culoarea fundalului paginii folosim:*

a) `<BODY background="culoare">`
b) `<BODY color="culoare">`
c) `<BODY bgcolor="culoare">`

16. *Setarea culorii pentru tot textul din pagină se face cu:*

a) `<FONT color="culoare">`
b) `<BODY fontcolor="culoare">`
c) `<BODY text="culoare">`

17. *Cum se poate schimba culoarea unui bloc de text?*

a) `<COLOR="culoare">text</COLOR>`
b) `<TEXT color="culoare">text</TEXT>`
c) `<FONT color="culoare">text</FONT>`

18. *Care sunt culorile prestabilite pentru legături, legături active şi legături vizitate?*

a) albastru, roşu, violet
b) albastru, verde, galben
c) negru, albastru, roşu

19. *Culoarea legăturilor se poate schimba folosind:*

a) `<LINK color="culoare">`
b) `<BODY link="culoare">`
c) `<BODY linkcolor="culoare">`

20. *Atributele* `link`**,** `alink` *şi* `vlink` *servesc la:*

a) stabilirea legăturilor
b) stabilirea culorilor pentru legături
c) stabilirea culorilor pentru întreaga pagină

Culori

1. a)
2. b)
3. b)
4. a)
5. c)
6. c)
7. c)
8. a)
9. b)
10. b)

Capitolul 4
Fonturi

Înainte de a aborda tipurile de caractere care pot fi utilizate în documentele Web, se cuvine menţionat un fapt: HTML nu este un limbaj orientat spre aspectul (*layout*) paginii, ci spre conţinutul acesteia.
Importanţa acestei menţiuni rezidă în aceea că etichetele HTML nu impun, cum s-ar putea crede, ci "sfătuiesc" marea varietate de browsere care procesează pagina Web, cum să afişeze textul. Ceea ce impun ele cu adevărat este conţinutul, textul în sine, nu forma de prezentare.
Desigur că nu se poate vorbi de cealaltă extremă, a unui arbitrariu absolut, în care fiecare browser va afişa textul sub o formă absolut imprevizibilă. Un text cu o anumită formatare pentru Internet Explorer, să spunem, va fi vizualizat practic identic de toţi utilizatorii acestui tip de browser.
Există numeroase etichete care permit formatarea caracterelor şi ne oferă posibilitatea de a da textului din pagina Web aspectul dorit.

1. Formatarea caracterelor

Eticheta care se foloseşte pentru a da unui caracter sau unui şir de caractere (unui text) aspectul dorit este eticheta container **<FONT> </FONT>.**
Între eticheta de început şi cea de sfârşit se inserează textul sau caracterul ale cărui caracteristici dorim să le stabilim.
Înainte de a discuta despre atributele etichetei <FONT> să luam un exemplu:

```
<FONT size="2" color="red" face="arial">Ion Luca Caragiale </FONT>
```

Efectul liniei de mai sus este afişarea textului "Ion Luca Caragiale" cu fonturi Arial, de culoare roşie şi de mărime egală cu 2 puncte.

Să analizăm succesiv cele trei atribute ale etichetei <FONT>:

size - dimensiunea textului - poate fi un număr între 1 şi 7 (dimensiuni absolute), între -1 şi -3 sau între +1 şi +4 (dimensiuni relative, adică raportate la dimensiunea deja existentă a fontului). Dimensiunea prestabilită (*default*) a fonturilor este 3.
Dacă dimensiunile menţionate depăşesc intervalul 1-7, browserul rotunjeşte automat valorile pentru a le încadra în acest interval.

color - culoarea textului - este culoarea cu care va fi afişat textul prezent între etichete (în exemplul nostru, "Ion Luca Caragiale").

face - tipul de font - determină tipul de font care va fi utilizat la afişarea textului. Tipurile cele mai uzuale sunt:

- Arial
- Tahoma
- Helvetica
- Times New Roman
- Courier
- Verdana

Atributul etichetei permite precizarea fie a unui singur tip de font fie a mai multora, separate prin virgulă. Precizarea mai multor fonturi ajută în eventualitatea când primul tip indicat nu este instalat pe calculatorul celui care vizitează pagina, şi se face ca în exemplul de mai jos.

```
<FONT face="arial, verdana, times new roman">
```

În această situaţie browserul va alege singur din listă primul tip de font pe care îl recunoaşte, ignorându-le pe următoarele.

Atenţie!

La stabilirea tipului de caracter utilizat în pagină ţineţi seama de faptul că nu toţi utilizatorii au instalate pe computerele lor acelaşi set de caractere ca şi dumneavoastră. Din acest motiv, folosirea unor caractere exotice, de efect, poate conduce la un eşec în cazul când utilizatorul nu are instalat acel font.
Din acest motiv este de preferat folosirea unor fonturi cu utilizare largă, cum sunt cele menţionate mai sus.

În Exemplul 4.1 vom aplica pe rând fiecare dintre cele trei atribute menţionate asupra unui text, aspectul paginii descrise de acest document fiind cel din Figura 4.1.

Exemplul 4. 1

```
<HTML>
  <HEAD>
    <TITLE>fonturi1</TITLE>
  </HEAD>
  <BODY bgcolor="yellow">
    <FONT size="5" face="tahoma" color="green">Manual de HTML si design
    Web</FONT><BR>
    <FONT size="+3" face="arial" color="red">Manual de HTML si design
    Web</FONT><BR>
    <FONT size="-1" face="courier new" color="blue">Manual de HTML si
    design Web</FONT>
  </BODY>
</HTML>
```

Pentru a stabili aspectul textului unei întregi pagini Web, putem folosi eticheta **<BASEFONT>.** Spre deosebire de <FONT> aceasta nu este o etichetă container, deoarece efectul ei se referă la tot textul din pagină. Nu se foloseşte pentru a stabili caracteristicile unui bloc de text.

Este indicat să fie inclusă în documentul HTML imediat după eticheta <BODY>. Atributele etichetei <BASEFONT> sunt aceleaşi cu cele ale etichetei <FONT>, respectiv: **size, color, face.**

În Exemplul 4.2 vom construi o pagină căreia îi setăm textul cu atributele: face - tip de font Arial, size - dimensiunea fontului 2, color - culoare albastră. O parte din text va avea alt tip de font, altă mărime şi culoare, acestea fiind stabilite prin folosirea etichetei <FONT>.

Exemplul 4. 2

```
<HTML>
  <HEAD>
    <TITLE>fonturi2</TITLE>
  </HEAD>
  <BODY bgcolor="white">
    <BASEFONT face="arial" color="blue" size="2">
      Am ajuns la lectia despre fonturi a <FONT face="arial black"
      color="red" size="4">Manualului de HTML</FONT>
  </BODY>
</HTML>
```

După cum puteţi observa din Figura 4.2, caracteristicile textului din pagină au fost setate cu ajutorul etichetei <BASEFONT>. Pentru a afişa simultan o parte din text într-un mod diferit am folosit eticheta <FONT>.

Deşi eticheta <BASEFONT> face parte din standardul HTML 3.2, atributele sale nu sunt recunoscute de browserele Netscape. Ca urmare, acesta va afişa textul utilizând propriile setări prestabilite (*default*), ignorând atributele menţionate în eticheta <BASEFONT>.

Următorul exemplu (Exemplul 4.3) afişează un cuvânt având literele de mărimi diferite. Aspectul acestei pagini este cel din Figura 4.3.

Exemplul 4. 3

```
<HTML>
  <HEAD>
    <TITLE>fonturi3</TITLE>
  </HEAD>
  <BODY bgcolor="white">
```

```
    <BASEFONT face="arial black" color="red">
    <FONT size="4">G</FONT>
    <FONT size="5">R</FONT>
    <FONT size="6">A</FONT>
    <FONT size="7">T</FONT>
    <FONT size="6">U</FONT>
    <FONT size="5">I</FONT>
    <FONT size="4">T</FONT>
  </BODY>
</HTML>
```

Să trecem în revistă, în continuare, alte etichete care schimbă aspectul unui caracter sau al unui bloc de text.

2. Accentuarea textului

1. Eticheta **<BIG> </BIG>** face fonturile mai mari decât dimensiunea curentă.

   ```
   <BIG>Text cu caractere mari</BIG>
   ```
   Text cu caractere mari

2. Eticheta **<SMALL> </SMALL>** face fonturile mai mici decât dimensiunea curentă.

   ```
   <SMALL>Text cu caractere mici</SMALL>
   ```
   Text cu caractere mici

3. Etichetele **<B> </B>** (*bold*) şi **<STRONG> </STRONG>** realizează scrierea cu caractere aldine, sau îngroşate (*bold*).

   ```
   <B>Text ingrosat 1</B>
   ```
   Text ingrosat 1

   ```
   <STRONG>Text ingrosat 2</STRONG>
   ```
   Text ingrosat 2

4. Etichetele **<I> </I>** (*italic*) şi **<EM> </EM>** (*emphasized*)realizează scrierea cu caractere italice.

   ```
   <I>Text inclinat 1</I>
   ```
   Text inclinat 1

   ```
   <EM> Text inclinat 2</EM>
   ```
   Text inclinat 2

5. Etichetele **<S> </S>** şi **<STRIKE> </STRIKE>** realizează scrierea textului tăiat de o linie orizontală.

   ```
   <S>Text taiat 1</S>
   ```
   ~~Text taiat 1~~

   ```
   <STRIKE>Text taiat 2</STRIKE>
   ```
   ~~Text taiat 2~~

6. Eticheta **<U> </U>** (*underlined*) realizează sublinierea textului.

   ```
   <U>Text subliniat</U>
   ```
   Text subliniat

Atenţie!

Utilizarea textului subliniat trebuie făcută cu grijă, întrucât acesta poate fi confundat cu un link.

7. Eticheta **** (*superscript*). Prin intermediul ei se pot insera în document caractere/texte plasate deasupra nivelului liniei de scriere.

   ```
   Apa ingheata la 0 <SUP>0</SUP>C
   ```
   Apa ingheata la 0 ^{0}C

8. Eticheta **** (*subscript*). Prin intermediul ei se inserează texte plasate sub nivelul liniei de scriere.

   ```
   Coordonatele X<SUB>1</SUB>, Y<SUB>2</SUB>
   ```
   Coordonatele X_1, Y_2

Este de remarcat faptul că etichetele <BIG> şi <SMALL> sunt executate diferit în diverse browsere.
Astfel, în Netscape <BIG> face textul mai mare cu un punct iar <SMALL> mai mic cu un punct decât dimensiunea curenta.
În Internet Explorer, <BIG> afişează textul cu fonturi de dimensiune 4 iar <SMALL> cu fonturi de dimensiune 2. Etichetele <BIG> şi <SMALL> pot fi repetate pentru a obţine un efect mai accentuat.

Exemplele următoare ilustrează modul cum pot fi utilizate aceste etichete.

Exemplul 4. 4

```
<HTML>
  <HEAD>
    <TITLE>fonturi4</TITLE>
  </HEAD>
  <BODY>
    <BASEFONT face="arial" color="blue">
    <BIG>Etichete</BIG> care <EM>schimba</EM> aspectul
    <SMALL>textului</SMALL>
  </BODY>
</HTML>
```

Aspectul paginii este cel din Figura 4.4

Exemplul 4. 5

```
<HTML>
  <HEAD>
    <TITLE>fonturi5</TITLE>
  </HEAD>
  <BODY>
    <FONT face="arial" size="5">Acesta este font Arial </FONT><BR>
    <FONT face="algerian" size="4" color="green">Acesta este font Algerian
    </FONT><BR>
    <FONT face="courier" color="blue"><STRONG>Acesta este font Courier
    </STRONG></FONT><BR>
    <FONT FACE="vivaldi" size="5" color="gray"><U>Acesta este font Vivaldi
    </U></FONT><BR>
    <FONT FACE="garamond"><EM>Acesta este font Garamond </EM></FONT><BR>
    <FONT FACE="modern" size="7" color="brown"><STRIKE>Acesta este font
    Modern </STRIKE></FONT><BR>
  </BODY>
</HTML>
```

Pagina arată ca în Figura 4.5

După cum aţi observat mai sus, există două etichete al căror efect este acelaşi: accentuarea textului fie prin scrierea cu caractere italice fie cu caractere aldine.

3. Etichete logice şi etichete fizice

Etichetele <B> şi <I> se numesc etichete **fizice**, iar <EM> şi <STRONG> etichete **logice**. Etichetele de titluri de la <H1> la <H6> sunt, de asemenea, etichete logice. Diferenţa dintre cele două tipuri de etichete este legată de felul cum sunt ele executate de diversele tipuri de browsere.

Astfel, etichetele fizice impun browserului afişarea în formatul exact, precizat prin intermediul lor. Dacă browserul nu suportă acest format, etichetele sunt ignorate.

Spre deosebire de etichetele fizice, cele logice lasă browserului libertatea de a alege cel mai bun mod de a îndeplini comanda transmisă prin eticheta. De exemplu, efectul etichetei <EM> este, în cele mai multe browsere, scrierea textului cu caractere italice. Totuşi, dacă un anumit tip de browser nu suportă acest format de caractere, va accentua totuşi textul într-un alt mod, cel mai bun de care dispune.

Alte etichete logice utilizate în documentele HTML sunt:

<CITE> </CITE> (*citation*) Etichetă folosită pentru inserarea unui citat. De obicei, citatul este afişat cu caractere italice.

<CODE> </CODE> (*code*) Etichete folosite pentru inserarea în text a unor secvenţe de cod scrise într-un limbaj de programare. În general secvenţele de cod sunt afişate cu fonturi Courier.

<DFN> </DFN> (*definition*) Eticheta este folosită pentru a insera în text definiţia unor termeni. Este utilă la crearea indexului sau glosarului unui document.

<KBD> </KBD> (*keyboard*) Această etichetă logică se foloseşte pentru a indica un text care urmează a fi introdus de la tastatură, ca în următorul exemplu:

```
Pentru a parasi programul tastati <KBD>quit</KBD>
```

<SAMP> </SAMP> (*sample*) Este o etichetă logică folosită pentru a insera o mostră de text.

<TT> </TT> (*teletype*) Această etichetă se foloseşte pentru a afişa textul inclus între etichete cu fonturi *monospaţiate*. Tipul de font monospaţiat (cum este de exemplu Courier) are proprietatea că fiecărui caracter i se rezervă acelaşi număr de pixeli pe ecran. Astfel, caracterul "i" va avea alocat tot atâta spaţiu pe ecran ca şi caracterul "W":

```
"Courier" este un font monospaţiat.
```

Mai trebuie precizat şi faptul că utilizarea unora dintre etichetele de formatare a fonturilor aşa cum au fost prezentate ele în acest capitol tinde să fie înlocuită cu foile de stiluri (*Cascade Style Sheets*) despre care vom vorbi într-un capitol viitor.

4. Rezumat

Pentru a stabili aspectul unei anumite secvenţe de text aceasta este inclusă între etichetele <FONT> </FONT> conform sintaxei:

```
<FONT size="numar" color="#RGB sau nume_culoare" face="tip_font"> text
</FONT>
```

Pentru a seta aspectul textului în toată pagina Web se foloseşte eticheta <BASEFONT> conform sintaxei:

```
<BASEFONT size="numar" color="#RGB sau nume_culoare" face="tip_font">
```

Accentuarea textului se poate face folosind diverse etichete pentru scrierea cu caractere aldine, cu caractere italice, cu caractere mai mari sau mai mici decât dimensiunea curentă a fonturilor.

Etichetele sunt de două tipuri: fizice şi logice, etichetele fizice fiind orientate către aspectul textului iar cele logice către semnificaţia textului în cadrul documentului.

5. Test

21. *Pentru a stabili tipul de font pentru întreaga pagină Web se foloseşte eticheta:*

a) <DEFAULT>
b) <BASEFONT>
c) <TARGET>

22. *Pentru a stabili tipul de font cu care se face afişarea unui text se foloseşte atributul:*

a) character
b) type
c) face

23. *Unul dintre exemplele următoare este incorect. Care?*

a) <FONT face="arial">
b) <FONT face="arial, verdana">
c) <FONTface="verdana">

24. *Ce realizează următoarea etichetă?*

```
<FONT size="+1">
```

a) Afişează textul cu fonturi de mărime 1.
b) Afişează pe pagină textul "+1".
c) Afişează textul cu fonturi de dimensiune mai mare cu un punct decât dimensiunea curentă.

25. *În afara etichetei* <B> *ce altă etichetă se mai foloseşte pentru a scrie un text cu caractere aldine?*

a) <DARK>
b) <STRONG>
c) <BIG>

26. *Etichetele* <I> *şi* <EM> *realizează:*

a) scrierea textului cu caractere italice
b) inserarea unei imagini
c) scrierea textului cu caractere mai mici decât cele curente

27. *Ce este incorect în următorul exemplu?*

```
<FONT face="arial, verdana, times new roman",
size="4"><B>text</FONT></B>
```

a) Nu se pot preciza mai multe tipuri de fonturi ca valori pentru atributul `face`.
b) Tipurile specificate nu se pot scrie cu caractere aldine
c) Etichetele `<FONT>` şi `<B>` nu se închid corect.

28. *Prin ce se aseamănă etichetele* `<STRONG>` *şi* `<EM>`*?*

a) Sunt amândouă etichete logice.
b) Sunt amândouă etichete fizice.
c) Servesc amândouă la scrierea textului cu caractere italice.

29. *Care este greşeala în următoarea construcţie?*

```
<BASEFONT face="arial" size="5">text</BASEFONT>
```

a) `<BASEFONT>` nu este o eticheta container
b) `<BASEFONT>` nu suportă atributul `face`
c) `<BASEFONT>` nu suportă atributul `size`

30. *Ce efect are următoarea etichetă?*

```
<FONT="arial, verdana, times new roman">
```

a) Textul se afişează cu toate cele trei tipuri de fonturi.
b) Textul se afişează cu primul tip de font, dintre cele trei, pe care îl recunoaşte browserul.
c) Eticheta nu era nici un efect întrucât este incorectă.

Fonturi

1. b)
2. c)
3. c)
4. c)
5. b)
6. a)
7. c)
8. a)
9. a)
10. b)

Capitolul 5
Formatarea textului

Etichetele despre care vom vorbi în continuare nu se mai referă la particularităţile fonturilor ci la felul în care poate fi amplasat un text în cadrul paginii. Înainte de a discuta în amănunt despre etichetele care permit

formatarea textului, să facem câteva precizări privind felul cum sunt afişate textele.

După cum ştiţi, fişierele HTML sunt fişiere text. Aceasta înseamnă că atunci când salvaţi fişierul respectiv, el este salvat numai ca text, deci browserul nu poate recunoaşte şi executa decât strict acele comenzi prevăzute de etichetele HTML. Din acest motiv, trecerea la un rând nou, care în editorul de text se realizează prin apăsarea tastei ENTER, nu este recunoscută de browser.

Un alt aspect este plasarea în text a spaţiilor. Chiar dacă veţi introduce spaţii suplimentare între cuvinte atunci când scrieţi documentul HTML în editorul de text, browserul va afişa invariabil un singur spaţiu între două cuvinte.

Acesta este motivul pentru care fiecare dintre aceste elemente de formatare a textului are o etichetă proprie care indică browserului cum anume să facă afişarea.

1. Trecerea la un rând nou

Trecerea la un rând nou se realizează cu ajutorul etichetei **
** (de la *line break*). Eticheta <BR> nu este o etichetă container. Ea are rolul de a comunica browserului că texul care urmează după etichetă va fi afişat pe un rând nou aşa cum rezultă din Exemplul 5.1.

Exemplul 5. 1

```
<HTML>
  <HEAD>
    <TITLE>text1</TITLE>
  </HEAD>
  <BODY>
    Buna ziua<BR>Ma numesc Lucia<BR>Invat sa construiesc o pagina Web<BR>
  </BODY>
</HTML>
```

Aspectul paginii descrise în exemplul de mai sus este cel din Figura 5.1.

Eticheta <BR> admite atributul **clear** care poate lua valorile `left`, `right` sau `all`. În mod normal, la întâlnirea etichetei <BR> linia curentă de text este întreruptă şi se face trecerea la o linie nouă, ca şi la acţionarea tastei *Enter* într-un editor de text. Există însă situaţii când dorim ca linia să fie afişată lângă un anumit element (o imagine, sau un tabel, de exemplu) care blochează partea din stânga sau din dreapta a liniei.
În aceste cazuri putem folosi atributul `clear` al etichetei <BR> ca în Exemplul 5.2. Puteţi observa efectul atributului `clear` al etichetei
 în Figura 5.2.

Exemplul 5. 2

```
<HTML>
  <HEAD>
    <TITLE>text2</TITLE>
  </HEAD>
  <BODY>
    <IMG src="../Imagini/toad.jpg" align="left">
    Acest text va fi afisat intre imagine si marginea dreapta a
    documentului.
    <BR clear="left">Acest text va fi afisat sub imagine, aliniat stanga.
    Intre acest text si dreapta imaginii va exista un spatiu gol.
  </BODY>
</HTML>
```

2. Titlurile (Headings)

Utilizarea titlurilor este justificată de nevoia de a sublinia prin format structura logică a unui document, dar, nu în ultimul rând, şi de monotonia la care expune un text lung, lipsit de variaţie în aspect. Fireşte că un asemenea text sfârşeşte prin a obosi cititorul Web, obişnuit cu texte scurte şi concentrate.
Din acest motiv se recomandă "spargerea" textelor mai lungi şi organizarea lor în secvenţe mai scurte, marcate de titluri şi subtitluri, organizate ierarhic, pe nivele.

Titlurile dintr-un document HTML sunt marcate cu ajutorul etichetelor **<H1>**, **<H2>**, **<H3>**, **<H4>**, **<H5>**, **<H6>**. Ele sunt etichete container, deci necesită eticheta corespunzătoare de închidere.
Eticheta <H1> defineşte titlul de dimensiunea maximă, iar <H6> pe cel de dimensiune minimă.

Atenţie!

Deşi standardul HTML nu interzice utilizarea în orice ordine a titlurilor, este de preferat ca acestea să fie folosite în ordinea "normală", progresiv, fără a sări peste vreun nivel (de pildă de la H1 direct la H5), pentru a evita eventualele probleme la convertirea textului în alte tipuri de fişiere.

În Exemplul 5.3 vom folosi toate cele şase etichete pentru titluri pentru a face o comparaţie cu dimensiunea standard a textului. Puteţi observa efectul acestor etichete în Figura 5.3.

Exemplul 5. 3

```
<HTML>
  <HEAD>
    <TITLE>text3</TITLE>
  </HEAD>
  <BODY>
    <H1>Titlu H1</H1>Text normal
    <H2>Titlu H2</H2>Text normal
    <H3>Titlu H3</H3>Text normal
    <H4>Titlu H4</H4>Text normal
    <H5>Titlu H5</H5>Text normal
    <H6>Titlu H6</H6>Text normal
  </BODY>
</HTML>
```

Este de menţionat o proprietate interesantă a etichetelor de titluri, şi anume aceea că un text scris pe aceeaşi linie cu un titlu este afişat în pagină pe rândul următor titlului, lăsându-se o linie liberă între titlu şi text, deşi nu este prezentă nici una dintre etichetele <BR> sau <P>.

Etichetele de titlu acceptă atributul **align** cu valorile left, center şi right pentru alinierea titlului blocului de text la stânga (în mod prestabilit), în centru şi respectiv la dreapta. Figura 5.4 redă aspectul paginii descrie de documentul din Exemplul 5.4 care ilustrează modul de aliniere al titlurilor.

Exemplul 5. 4

```
<HTML>
  <HEAD>
    <TITLE>text4</TITLE>
  </HEAD>
  <BODY>
```

```
    <H1 align="center">Titluri</H1><HR>
    <H1 align="center">
      Titlu de marime 1 aliniat in centru
    </H1>
    <H2 align="right">
      Titlu de marime 2 aliniat la dreapta
    </H2>
    <H4>
      Titlu de marime 4 aliniat la stanga
    </H4>
  </BODY>
</HTML>
```

3. Paragrafele

Trecerea la un nou paragraf în cadrul textului se realizează cu ajutorul etichetei **<P> </P>**. Această etichetă comunică browserului să insereze o linie liberă în text şi să înceapă o nouă linie. Eticheta <P> este o etichetă container dar prezenţa etichetei de închidere </P> este opţională în anumite situaţii. Dacă ea este prezentă, indică browserului să insereze o linie liberă şi după blocul de text cuprins între cele două etichete.

Eticheta <P> admite atributul **align** cu cele trei valori ale acestuia: `left`, `center` şi `right`, care permit alinierea textului la stânga, centrat respectiv la dreapta. Exemplul 5.5 ilustrează cele trei modalităţi de aliniere a textului.

Exemplul 5. 5

```
<HTML>
  <HEAD>
    <TITLE>text5</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Paragrafe</H1><HR>
    <P> Acesta este un paragraf aliniat la stanga.
    <P align="right"> Acesta este un paragraf aliniat la dreapta.
    <P align="center"> Acesta este un paragraf aliniat la centru.
  </BODY>
</HTML>
```

Din Figura 5.5 care redă aspectul paginii descrise în exemplu se poate observa că alinierea la stânga este implicită (nu mai trebuie specificat `align="left"`). Se mai poate observa că nu a fost folosită eticheta de închidere a paragrafelor, deoarece la întâlnirea unei noi etichete <P> vechiul paragraf se consideră închis. Totuşi, prezenţa sau absenţa etichetei de închidere a paragrafului atunci când este prezent atributul `align` trebuie stabilită cu grijă, pentru a nu obţine rezultate neconforme cu dorinţele dumneavoastră.

În Exemplul 5.6 dorim să aliniem un nou paragraf în dreapta paginii, iar textul de pe rândul următor paragrafului, care este introdus prin eticheta <BR>, să fie scris normal, de la capătul din stânga al paginii.

Exemplul 5. 6

```
<HTML>
  <HEAD>
    <TITLE>text6</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Paragrafe</H1><HR>
    Acest text este scris normal, incepand e la marginea din stanga.
    <P align="right">Acest text este aliniat la dreapta
```

```
    <BR>Unde este afisat acest text?
  </BODY>
<HTML>
```

Din Figura 5.6 se poate observa că afişarea nu s-a făcut conform intenţiilor noastre. Motivul este faptul că eticheta
<P align=”right”> nu are eticheta de închidere </P>, şi prin urmare efectul său se prelungeşte până la întâlnirea unei alte etichete <P>.
Dacă veţi închide paragraful în mod explicit cu ajutorul etichetei de închidere, înainte de trecerea la o linie nouă, efectul asupra textului va fi cel dorit aşa cum rezultă din Figura 5.7.

Deşi aparent etichetele
 şi <P> au un efect asemănător, şi anume trecerea la o linie nouă, ele nu sunt executate la fel.
 comunică browserului unde se încheie o linie, în vreme ce <P> îi impune să lase o linie liberă şi să treacă la un nou paragraf. Dacă folosim atributul `align` al etichetei <P> în situaţiile când după paragraf este introdus un alt element în pagină este recomandată închiderea în mod explicit a etichetei respective.

4. Preformatarea textului

Aşa cum am precizat la începutul capitolului, atunci când editaţi documentul HTML cu ajutorul unui editor de text, indiferent câte spaţii veţi lăsa între cuvinte sau între liniile de text, browserul va afişa textul invariabil cu un singur spaţiu între cuvinte şi va ajusta lungimea liniilor în funcţie de mărimea ferestrei. Cu alte cuvinte, orice spaţii sau linii libere suplimentare vor fi ignorate de către browser. Aşa cum am văzut mai sus, trecerea la o linie nouă se realizează cu ajutorul etichetei <BR> iar inserarea unei linii libere se face cu ajutorul etichetei <P>.

Există, însă, şi o etichetă care impune browserului să afişeze textul întocmai cum a fost el formatat în documentul HTML. Este vorba despre eticheta **<PRE> </PRE>.**
Eticheta <PRE> permite preformatarea textului şi menţine spaţierea şi alinierea textului aşa cum a fost făcută în documentul sursă HTML. Eticheta de închidere </PRE> este obligatorie.

Exemplul 5.7 ilustrează modul cum poate fi folosită eticheta <PRE>. Aspectul paginii descrise în acest document este cel din Figura 5.8.
Testaţi exemplul folosind eticheta <PRE> iar apoi eliminaţi-o, pentru a observa mai bine efectul său.

Exemplul 5. 7

```
<HTML>
  <HEAD>
    <TITLE>text7</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Textul preformatat</H1><HR>
      Acesta este un text normal <P>
      Orar:

      Luni

      8:00 Romana
      9.00 Matematica
      10:00 Geografie
      11.00 Istorie
```

```
      12.00 Fizica<P>

      Acesta este textul de mai sus preformatat
    <PRE>
      Orar:
      Luni

      8:00 Romana
      9.00 Matematica
      10:00 Geografie
      11.00 Istorie
      12.00 Fizica
    </PRE>
  </BODY>
</HTML>
```

5. Centrarea textului

Aşa cum am văzut mai sus, afişarea textului centrat în pagină se poate face cu ajutorul etichetei <P> având atributul `align` setat la valoarea `"center"`.
O altă posibilitate de a realiza acest lucru este de a utiliza o etichetă dedicată, incluzând textul între etichetele
<CENTER> </CENTER>. Eticheta <CENTER> este o etichetă container, prezenţa etichetei de închidere fiind obligatorie.
Exemplul 5.8 realizează centrarea unui text.

Exemplul 5. 8

```
<HTML>
  <HEAD>
    <TITLE>text8</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Centrarea textului </H1><HR>
    <CENTER>
      Acesta este un text centrat. Acesta este un text centrat. Acesta
      este un text centrat. Acesta este un text centrat. Acesta este un
      text centrat. Acesta este un text centrat. Acesta este un text
      centrat. Acesta este un text centrat.
    </CENTER>
  </BODY>
</HTML>
```

Aspectul paginii descrise în exemplul de mai sus este cel din Figura 5.9.

6. Afişarea textului pe o singură linie

În cazul când se doreşte afişarea unui text pe o singură linie, chiar dacă acest lucru înseamnă depăşirea marginii ferestrei şi derularea acestuia pe orizontală, se include blocul de text între etichetele **<NOBR> </NOBR>**.
Această etichetă împiedică browserul să limiteze lungimea liniei la dimensiunea ferestrei.
Exemplul 5.9 ilustrează folosirea etichetei <NOBR>, pagina având aspectul din Figura 5.10.

Exemplul 5. 9

```
<HTML>
  <HEAD>
    <TITLE>text9</TITLE>
  </HEAD>
  <BODY>
```

```
    <H1 align="center">Textul pe o singura linie </H1><HR>
    <NOBR>
      Acesta este un text care va fi afisat pe o singura linie, chiar daca
      depaseste dimensiunile unei ferestre obisnuite. Puteti verifica
      acest fapt prin reducerea dimensiunilor ferestrei browserului.
    </NOBR>
  </BODY>
</HTML>
```

7. Blocul <DIV>

O altă modalitate de delimitare şi de formatare a unui bloc de text este folosirea etichetei container **<DIV> </DIV>**. Prezenţa etichetei de închidere este obligatorie.
Eticheta <DIV> realizează divizarea unui document HTML în secţiuni distincte, diviziune în care pot fi incluse, pe lângă text, şi alte elemente: legături, imagini, formulare. Ca şi în cazul etichetei pentru introducerea paragrafelor, eticheta <DIV> admite atributul **align**, pentru alinierea textului. Valorile posibile ale acestui atribut, deja cunoscute, sunt:

- `left` - aliniere la stânga
- `center` - aliniere la centru
- `right` - aliniere la dreapta

Alinierea precizată de atributul `align` al blocului are efect asupra tuturor elementelor incluse în blocul <DIV>. Blocul <DIV> admite şi atributul **nowrap**, care interzice întreruperea rândurilor de către browser.

În documentul descris în Exemplul 5.10 este ilustrată utilizarea acestei etichete. Pagina va avea aspectul redat în Figura 5.11.

Exemplul 5. 10

```
<HTML>
  <HEAD>
    <TITLE>text10</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Blocul div</H1><HR>
    Aceasta linie este o linie de text normala.
    <DIV align="right">
      Aceasta este prima sectiune a textului, aliniata dreapta.<BR>
    </DIV>
    <DIV align="center">
      Aceasta este a doua sectiune a textului, aliniata central.<BR>
    </DIV>
    <DIV align="left">
      Aceasta este a treia sectiune a textului, aliniata stanga.<BR>
    </DIV>
  </BODY>
</HTML>
```

8. Linii orizontale

Într-o pagina Web pot fi inserate linii orizontale care au rolul de a delimita diferitele zone ale paginii sau de a îmbunătăţi aspectul ei estetic. Acest lucru se face cu ajutorul etichetei **<HR>.** Eticheta <HR> nu este o etichetă container deci nu există o etichetă de închidere.
Pentru a configura o linie orizontală se utilizează următoarele atribute ale etichetei <HR>:

align - permite alinierea liniei orizontale. Valorile posibile sunt `left`, `center` şi `right`
width - specifică lungimea liniei - lungimea poate fi stabilită în pixeli sau în procente din lăţimea ecranului.
size - specifică grosimea liniei, exprimată în pixeli
color - permite definirea culorii liniei

Atributele etichetei `<HR>` sunt ilustrate în Exemplul 5.11.

Exemplul 5. 11

```
<HTML>
  <HEAD>
    <TITLE>text11</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center"> Linii orizontale </H1>
    Linie aliniata la stanga, lungime 100%, grosime 2 <HR>
    Linie aliniata in centru , lungime 50%, grosime 5 pixeli.
    <HR align="center" width="50%" size="5" color="black">
    Linie aliniata la dreapta , lungime 150 de pixeli, grosime 12 pixeli ,
    de culoare rosie.
    <HR align="right" width=150 size=12 color="red">
  </BODY>
</HTML>
```

Din Figura 5.12 care redă aspectul paginii descrise în exemplul anterior se poate observa că simpla prezenţă a etichetei `<HR>` fără nici un fel de atribute duce la afişarea unei linii predefinite, de lungime egală cu 100% din pagină şi grosimea egală cu 2 pixeli.

Atributul `color` al etichetei `<HR>` nu este suportat de browserul Netscape.

În exemplul următor (Exemplul 5.12) vom ilustra modul în care eticheta `<DIV>` aliniază elementele conţinute în interiorul său, în cazul nostru text şi linii orizontale. Pagina descrisă în acest exemplu va avea aspectul din Figura 5.13.

Exemplul 5. 12

```
<HTML>
  <HEAD>
    <TITLE>text12</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Linii orizontale</H1><HR>
    Linia de mai jos este aliniata la stanga
    <HR size="3" color="blue" width="40%" align="left">
    Liniile de mai jos sunt aliniate la centru cu ajutorul blocului DIV
    <DIV align="center">
      Linii aliniate la centru
      <HR size="10" color="cyan" width="50%">
      <HR size="5" color="navy" width="400">
    </DIV>
  </BODY>
<HTML>
```

9. Inserarea unei adrese poştale

Dacă într-o pagină Web trebuie inclusă o adresă poştală, atunci putem folosi eticheta următoare: **<ADDRESS> </ADDRESS>**.

Eticheta <ADDRESS> este o etichetă logică şi are drept efect, în cele mai multe browsere, afişarea textului cu caractere italice.

În Exemplul 5.13 este inserată în pagină o adresă, aspectul paginii fiind cel din Figura 5.14.

Exemplul 5. 13

```
<HTML>
  <HEAD>
    <TITLE>text13</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center"> Adresa </H1><HR>
    Adresa firmei noastre este
    <ADDRESS>
      GoldenWeb Consult <BR>
      Str. Paradisului, Nr. 1<BR>
      Bucuresti
    </ADDRESS>
  </BODY>
</HTML>
```

10. Inserarea unui citat

Pentru a insera un citat într-un text se pot folosi două etichete, în funcţie de dimensiunea citatului.
Astfel, dacă citatul depăşeşte câteva linii, se foloseşte eticheta **<BLOCKQUOTE> </BLOCKQUOTE>.**
Efectul acestei etichete este afişarea indentată a textului cuprins între eticheta de început şi cea de sfârşit (în majoritatea browserelor) sau afişarea acestuia cu caractere italice (rareori).

Dacă dorim ca într-un text să citam un termen, titlul unei lucrări sau să dăm o referinţă, putem include textul respectiv între etichetele **<CITE> </CITE>.**
În cele mai multe dintre browsere, textul inclus între etichetele <CITE> va fi afişat cu caractere italice.
Exemplul 5.14 ilustrează modul de folosire al celor două etichete. Pagina descrisă în exemplu are aspectul din Figura 5.15.

Exemplul 5. 14

```
<HTML>
  <HEAD>
    <TITLE>text14</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Inserarea unui citat</H1><HR>
    Textul de mai jos este un citat
    <BLOCKQUOTE>
      Probabilitatea ca o felie de paine unsa cu unt sa cada pe covor cu
      fata unsa in jos este direct proportionala cu valoarea covorului
    </BLOCKQUOTE>
    Citatul de mai sus face parte din <CITE>Legile lui Murphy</CITE>
  </BODY>
</HTML>
```

11. Inserarea caracterelor speciale

Deşi este impropriu să numim caracterul "blank" sau *space* un caracter special, având în vedere frecvenţa cu care este folosit, totuşi acesta, ca şi alte caractere,

este inserat în pagină cu ajutorul unei comenzi speciale: comanda **&** (comanda ampersand).

Aşa cum ştiţi deja, indiferent câte spaţii vom introduce atunci când editǎm documentul HTML, browserul le va ignora şi va afişa cuvintele cu un singur spaţiu între ele. Pentru a forţa introducerea spaţiilor suplimentare se foloseşte comanda ` ` (*no break space*).

Atenţie

Comanda începe cu simbolul **&** (ampersand) şi se termină cu **;** (punct şi virgulă).

În Exemplul 5.15 este ilustrat modul cum pot fi spaţiate cuvintele folosind comanda ` `

Exemplul 5. 15

```
<HTML>
  <HEAD>
    <TITLE>text15</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Inserarea caracterelor speciale</H1><HR>
    <FONT size="2" face="arial">
      Cuvintele      din      acest 
          text      sunt     
      despartite      de      grupuri 
          de      trei     
      spatii.
    </FONT>
  </BODY>
</HTML>
```

După cum observaţi din Figura 5.16 cuvintele sunt despărţite prin trei spaţii în loc de unul singur. Puteţi adăuga oricâte spaţii doriţi în acest mod. De asemenea, pentru a indenta un text, puteţi insera la începutul fiecărei linii numărul de comenzi ` ` egal cu numărul de spaţii cu care vreţi să indentaţi textul.

Iată mai jos un tabel cu comenzile necesare pentru a insera în text cele mai cunoscute caractere.

Tabelul 5. 1

Denumire caracter	Reprezentarea grafică	Comanda HTML
Spaţiu liber		` `
Copyright	©	`©`
Trademark	™	`™`
Registered	®	`®` sau `®`
Mai mic decât	<	`<`
Mai mare decât	>	`>`
Ampersand	&	`&`

Ghilimele	"	"
Cent	¢	&#162
Un sfert	¼	&#188
O jumătate	½	&#189
Trei sferturi	¾	&#190
Grade	°	&#176

12. Sugestii privind aspectul textului

Aspectul textului într-o pagină Web este esenţial pentru calitatea acesteia. Un text lizibil, scris cu caractere de dimensiune potrivită, bine organizat şi aşezat în pagină va face din pagina dumneavoastră un mediu accesibil şi prietenos care îl va stimula pe vizitatorul acesteia să parcurgă cu plăcere informaţiile oferite. Iată câteva sfaturi referitoare la aspectul paginilor Web, menite să vă ajute la crearea unor texte lizibile, cât mai uşor de parcurs de către cititori.

- Nu folosiţi fonturi de dimensiuni prea mici sau fonturi cu serife, cum este Times New Roman. Deşi este tipul cel mai utilizat pentru publicaţiile tipărite, textele scrise cu Times New Roman sunt mai greu de citit de pe monitor. Preferaţi tipurile Arial, Verdana, Helvetica.

- Nu folosiţi numai litere mari (majuscule). Textele scrise cu majuscule sunt mai greu de citit.

- Nu faceţi exces de caractere aldine (îngroşate, *bold*). Limitaţi-vă la a accentua cuvinte sau porţiuni reduse din text. De asemenea, nu folosiţi caractere cursive (înclinate, sau italice) în exces.

- Nu folosiţi prea multe tipuri de fonturi şi nici tipuri de fonturi rare. Folosirea unui mare număr de fonturi de dimensiuni şi culori diferite va face ca paginile să arate încărcate şi să fie greu de citit. Folosirea tipurilor de fonturi rare aduce după sine riscul ca vizitatorul să nu le aibă instalate pe computerul propriu, astfel că textul va fi afişat de browser cu fonturile prestabilite.

- Nu aranjaţi textul în pagină pe două coloane. Aceasta este o practică obişnuită în ziare şi reviste dar nu este potrivită într-o pagină Web deoarece forţează vizitatorul ca odată ajuns la baza paginii să o deruleze din nou pentru a citi ce-a de-a două coloană.

- Este bine să folosiţi fiecare etichetă în scopul pentru care a fost ea definită. Nu folosiţi etichetele de titlu pentru a accentua anumite părţi ale textului care nu sunt titluri. Nu folosiţi eticheta `<ADDRESS>` pentru a insera alte texte decât adrese sau eticheta `<CITE>` pentru a realiza scrierea cu caractere italice a textului.

Motoarele de căutare indexează paginile Web şi în funcţie de elementele (etichetele) care sunt incluse în ele iar folosirea lor neadecvată conduce la o indexare eronată a paginii dumneavoastră.
Multe motoare de căutare scanează pagina în căutarea titlurilor şi pe baza lor construiesc o schiţă a paginii.
Dacă veţi folosi etichetele de titlu pentru a accentua anumite părţi ale textului (operaţie pe care o puteţi realiza în mod corect folosind eticheta `<FONT>`) veţi

oferi informaţii eronate motoarelor de căutare. În acest fel, scad şansele ca paginile dumneavoastră să fie identificate şi citite de cei interesaţi.

Atenţie!

Nu uitaţi că paginile vor fi publicate pe Web, unde, pentru a-şi atinge scopul, adică pentru a fi citite, trebuie mai întâi descoperite de cititori, cel mai frecvent prin intermediul motoarelor de căutare.

13. Rezumat

Browserul afişează textele cu câte un singur spaţiu între cuvinte şi nu recunoaşte sfârşitul de linie şi începutul unei linii noi. Elementele de formatare a textului se introduc în documentul HTML prin intermediul unor etichete specifice:

- trecerea la un rând nou: eticheta

  ```
  <BR clear="left, right, all">text
  ```
- introducerea unui titlu: etichetele <H1> - <H6>
  ```
  <Hn align="left, right, center">text</Hn>
  ```
- introducerea unui paragraf: eticheta <P> </P>
  ```
  <P align="left, right, center" nowrap>text</P>
  ```
- preformatarea textului: eticheta <PRE> </PRE>
  ```
  <PRE>text</PRE>
  ```
- centrarea textului: eticheta <CENTER> </CENTER>
  ```
  <CENTER>text</CENTER>
  ```
- afişarea textului pe o singură line: eticheta <NOBR> </NOBR>
  ```
  <NOBR>text</NOBR>
  ```
- crearea unui diviziuni în text: eticheta <DIV> </DIV>
  ```
  <DIV align="left, right, center">text</DIV>
  ```
- inserarea unei linii orizontale: eticheta <HR>
  ```
  <HR size="valoare" width="valoare" color="#RGB sau nume_culoare"
  align="left, right, center">
  ```
- inserarea unei adrese: eticheta <ADDRESS> </ADDRESS>
  ```
  <ADDRESS>text</ADDRESS>
  ```
- inserarea unui citat: etichetele <BLOCKQUOTE> </BLOCKQUOTE> şi <CITE> </CITE>
  ```
  <BLOCKQUOTE>text</BLOCKQUOTE>
  <CITE>text</CITE>
  ```
- inserarea caracterelor speciale prin comanda **&;**

14. Test

31. *Care este rolul etichetei*
?

a) Trecerea la un rând nou
b) Inserarea unei linii libere în text
c) Inserarea unei linii orizontale

32. *Pentru ca browserul să afişeze textul aşa cum a fost el formatat în documentul HTML se foloseşte eticheta:*

a) `<P>`
b) `<PRE>`
c) `<DIV>`

33. *Ce se va afişa în următorul exemplu?*

```
<P>Text1
<P align="right">Text2
```

a) `Text1` şi `Text2` vor fi aliniate la dreapta, cu o linie liberă între ele
b) `Text1` va fi aliniat la stânga, `Text2` va fi aliniat la dreapta, cu o linie liberă între ele
c) `Text1` şi `Text2` vor fi scrise pe aceeaşi linie

34. *Pentru a scrie un titlu de dimensiune maximă folosim eticheta:*

a) `<H1>`
b) `<H6>`
c) `<H7>`

35. *Pentru a afişa un text pe o singura linie:*

a) folosim eticheta `<BR>`
b) folosim eticheta `<NOBR>`
c) nu folosim nici o etichetă

36. *Care dintre următoarele etichete nu realizează centrarea textului?*

a) `<P align="center">Text</P>`
b) `<CENTER>Text</CENTER>`
c) `<DIV>Text</DIV>`

37. *Ce realizează eticheta* `<ADDRESS>`*?*

a) inserarea unui link
b) inserarea unei adrese de e-mail
c) inserarea unei adrese poştale

38. *Pentru a insera într-o pagină o linie orizontală standard folosim eticheta:*

a) `<HR>`
b) `<HR width="100">`
c) `<HR size="1">`

39. *Eticheta* `<BLOCKQUOTE>` *foloseşte la:*

a) indentarea textului
b) scrierea textului cu caractere italice
c) inserarea unui citat în text

40. *Care dintre aceste afirmaţii este falsă?*

a) Toate spaţiile şi liniile libere introduse în text la editarea documentului HTML sunt afişate întocmai de browser.

b) Un text scris pe aceeaşi linie cu o etichetă de titlu este afişat sub titlu, chiar dacă nu este prezentă eticheta <BR>.
c) Pentru a introduce spaţii suplimentare în text se foloseşte comanda

Formatarea textului

1. a)
2. b)
3. b)
4. a)
5. b)
6. c)
7. c)
8. a)
9. c)
10. a)

Capitolul 6
Legături (link-uri)

Legăturile (*link*-urile) reprezintă, poate, partea cea mai importantă a unei pagini Web. Ele transformă un text obişnuit în hypertext - un nou tip de text, diferit de cel din clasicele pagini de carte, care permite trecerea rapidă de la o informaţie aflată pe un anumit computer la altă informaţie memorată pe un alt computer localizat oriunde în lume.

1. Adresa URL

Pentru a stabili o legătură cu o altă pagină (un alt fişier) trebuie să specificăm **adresa URL** a acestuia. URL este un acronim, de la *Uniform Resource Locator*, şi reprezintă adresa de identificare a unei resurse (a unui fişier) aflată pe calculatorul propriu sau pe orice alt calculator din lume conectat la Internet. O adresă URL constă dintr-un şir de caractere care identifică în mod unic o anumită resursă, oferind informaţii despre numele serverului pe care este stocată acea resursă precum şi despre localizarea ei.

Există mai multe etichete HTML care folosesc adresa URL drept valoare pentru anumite atribute: etichetele care introduc legături, imagini sau formulare. Toate acestea folosesc aceeaşi sintaxă a adresei URL pentru a specifica locaţia unei anumite resurse, indiferent de tipul acesteia.

Sintaxa generală a unei adrese URL este:

schema://server_gazda:port/calea_catre_fisier

unde:
- **schema** - reprezintă unul dintre protocoalele de transfer al informaţiilor, cum ar fi HTTP, FTP, Gopher, Telnet, etc.
- **server_gazda** - reprezintă identificatorul serverului pe care este găzduit fişierul respectiv. Acest identificator poate fi adresa IP a serverului sau numele său.
- **port** - reprezintă numărul portului de comunicaţie prin intermediul căruia browserul se conectează la server. Serverele deţin mai multe astfel de porturi, fiecare servind unui alt tip de comunicaţie: HTTP, FTP, poştă electronică, etc. Portul prestabilit în cazul transferului prin HTTP are numărul 80. Numărul portului trebuie precizat numai în cazul când acesta este diferit de 80.

- **calea_catre_fisier** - reprezintă localizarea ierarhică a fişierului în sistemul de directoare de pe server. Aceasta constă într-unul sau mai multe nume despărţite prin caracterul "/" (*slash*)

Iată câteva exemple de adrese URL împreună cu explicaţii referitoare la sintaxa lor:

http://www.autor.com/carti.html
În acest exemplu adresa URL indică documentul HTML numit *carti.html* care se află directorul rădăcină al serverului www.autor.com

http://www.autor.com/
Această adresă indică prima pagină (*home page*) a aceluiaşi server.

http://www.autor.com:8080/
Exemplul de mai sus indică de asemenea spre prima pagină a serverului www.autor.com, dar specificând şi numărul portului care este diferit de cel prestabilit.

http://www.autor.com/carti.html#webdesign
În acest exemplu este indicată calea spre documentul HTML *carti.html* aflat pe acelaşi server, dar specificându-se o anumită secţiune a acestui document, secţiune denumită *webdesign*.

În cazul când fişierul spre care este făcută legătura este stocat pe propriul calculator se foloseşte o adresă URL de forma următoare:

file://server/calea_catre_fisier

unde:
- **server** - reprezintă numele computerului pe care este stocat fişierul (acelaşi pe care rulează browserul). În acest caz, browserul va accesa fişierul folosind facilităţile obişnuite ale sistemului de operare de pe computerul propriu.
- **calea_catre_fisier** - reprezintă localizarea fişierului în structura de directoare conform regulilor sistemului de operare

Numele computerului personal poate fi înlocuit cu numele generic **"localhost".** În acest caz, adresa URL poate avea următoarea formă:

file://localhost/calea_catre_fisier

De exemplu fişierul *culori.html* care este salvat în folderul de lucru *Work* din *My Documents* pe computerul personal va avea următoarea adresă URL:

file://localhost/C:\My Documents\Work\culori.html

Termenul *localhost* poate fi omis, deoarece este numele prestabilit al computerului propriu. În acest caz este necesară prezenţa unui caracter "/" suplimentar:

file:///C:\My Documents\Work\culori.html

În practică, atunci când dorim să adresăm un fişier aflat pe computerul propriu putem omite prima parte a adresei URL obţinând următoarea formă:

C:\My Documents\Work\culori.html

Atenţie!

Se poate observa că în cazul resurselor stocate pe servere externe, numele directoarelor care fac parte din calea către resursă sunt despărţite prin caracterul "/" (*slash*).
La fişierele stocate pe computerul propriu, directoarele şi subdirectoarele care formează calea către fişier sunt despărţite prin caracterul "\" (*backslash*). Cele două moduri de scriere sunt specifice celor două sisteme de operare: UNIX, respectiv DOS-Windows.

Deoarece marea majoritate a serverelor Web folosesc sistemul de operare UNIX, calea către resursele stocate pe ele se scrie respectând convenţiile acestui sistem. Mai mult, această convenţie s-a extins şi la adresele URL referite de pe sisteme Windows. Astfel, dacă doriţi să adresaţi un fişier numit *foto.jpg* aflat pe discul *C:*, în *My Documents*, folderul *Imagini* puteţi folosi adresa URL:

C:/My Documents/Imagini/foto.jpg

2. Adrese absolute şi adrese relative

Pentru a putea localiza un fişier în structura ierarhică de directoare, în scopul de a stabili o legătură către el, se poate folosi adresarea **absolută** sau adresarea **relativă**.

Adresa absolută a unui fişier conţine calea precisă şi completă către fişierul respectiv pornind de la vârful ierarhiei de directoare:

C:/Manual HTML/Exemple/culori.html

Fişierul *culori.html* se află plasat pe discul *C:*, în directorul *Manual HTML*, subdirectorul *Exemple*.

Adresa relativă a unui fişier precizează poziţia acestuia în raport cu documentul HTML din care este apelat. Vom reveni ceva mai jos asupra acestui subiect.

3. Stabilirea legăturilor

Pentru a insera legături într-un document HTML folosim eticheta **<A> </A>.** Eticheta <A> este o etichetă container, prezenţa etichetei de închidere fiind obligatorie.

Atributul obligatoriu al etichetei <A> este **href** (*Hypertext Reference*) care primeşte ca valoare adresa URL a fişierului cu care dorim să stabilim legătura. Acest fişier poate fi un document HTML, o imagine sau un fişier de alt tip. Documentul HTML în care este prezentă legătura se numeşte **sursă** iar fişierul către care este făcută legătura se numeşte **ţintă**. Sintaxa etichetei <A> este următoarea:

```
<A href="adresa_URL">text sau imagine</A>
```

Între etichetele <A> şi </A> poate fi plasat un text obişnuit sau o imagine. În mod prestabilit textul inclus între etichetele <A> este afişat subliniat şi de culoare albastră iar imaginile au un chenar de culoare albastră. Folosirea etichetei <A> imbricată cu etichete de formatare a textului, fonturi, liste sau tabele se face plasând eticheta <A> *în interiorul* acestora.
În acest sens, standardul HTML consideră *incorectă* o construcţie ca aceasta:

```
<A href="adresa_URL"><FONT face="tip">Legatura</FONT></A>
```

În locul ei vom folosi construcţia:

```
<FONT face="tip"><A href="adresa_URL">Legatura</A></FONT>
```

Atenţie!

Nu este permisă imbricarea mai multor etichete <A>.

În funcţie de localizarea ţintei (a resursei referite de legătură) putem clasifica legăturile astfel:

- legături în cadrul aceleiaşi pagini (ancore)
- legături către o pagină aflată în acelaşi folder
- legături către o pagină aflată în alt folder
- legături către pagini externe

4. Ancore - legături în cadrul aceleiaşi pagini

Pentru a facilita navigarea într-o pagină care conţine un text de mari dimensiuni se pot insera în acesta anumite puncte de reper către care se definesc legături. Acestea se numesc **ancore**.

Pentru a plasa o ancoră sunt necesare două elemente:

1. **Punctul** spre care dorim să facem legătura.
   Acesta se defineşte inserând în punctul din pagină dorit (de obicei în dreptul unui anumit element din pagină: un titlu, o imagine, o altă legătură, un tabel, etc.) eticheta <A>, însoţită de atributul **name** care primeşte ca valoare un nume de identificare atribuit ancorei (de exemplu "nume_ancora"). Prin urmare, identificarea punctului spre care se face legătura se realizează astfel:

```
<A name="nume_ancora"> </A>
```

2. **Legătura propriu-zisă,** care se defineşte folosind atributul **href** al etichetei <A>. În exemplul de mai sus, acesta primeşte ca valoare "#nume_ancora". Stabilirea legăturii se realizează după următoarea sintaxă:

```
<A href="#nume_ancora">text explicativ</A>
```

Textul explicativ va fi afişat în mod diferit, în format hyperlink, subliniat şi de culoare prestabilită albastră.

În momentul când se efectuează click cu mouse-ul pe text explicativ se realizează un salt în cadrul paginii, browserul afişând partea din pagină care începe de la elementul în dreptul căruia a fost inserată ancora.

Atenţie!

Prezenţa semnului **#**, plasat înaintea numelui ancorei, este obligatorie. Acesta indică browserului faptul că este vorba despre o legătură internă, în cadrul paginii. În cazul în care semnul este omis, browserul va căuta acest nume în afara paginii, unde, evident, nu îl va găsi.

Pentru a introduce o legătură către o ancoră definită în alt document (altă pagină) aflat în acelaşi director, atributul `href` primeşte o valoare de forma:

```
href="nume_fisier.html#nume_ancora".
```

Exemplul 6.1 ilustrează cele două situaţii. Pentru a exemplifica modul în care poate fi inserată o ancoră într-un alt document şi cum poate fi ea referită, am inserat în documentul *text14.html* ancora `<A name="citat">`.

Exemplul 6. 1

```
<HTML>
  <HEAD>
    <TITLE>legaturi1</TITLE>
  </HEAD>
  <BODY>
    <A name="ancora1"></A>
    <H1 align="center">Ancore definite in acelasi document</H1><HR>
    <BR>A<BR>B<BR>C<BR>D<BR>E
    <BR>F<BR>G<BR>H<BR>I<BR>J
    <BR>K<BR>L<BR>M<BR>N<BR>O
    <BR>P<BR>R<BR>S<BR>T<BR>U
    <BR>V<BR>W<BR>Z<BR>X<BR>
    <A href="#ancora1">Sus</A>
    <BR><BR>
    <H1 align="center" >Ancore definite in alt document</H1><HR><P>
    Click
    <A href="text14.html#citat">AICI </A>
    pentru a deschide un document situat in alta pagina
  </BODY>
</HTML>
```

Aspectul paginii descrise în acest exemplu este cel din Figura 6.1.

Observaţie

Construcţiile de mai jos au acelaşi rol, şi anume inserarea unei ancore denumită `"ancora1"` în punctul din pagină care conţine elementul "ELEMENT".

```
<A name="ancora1">ELEMENT</A>
<A name="ancora1"></A>ELEMENT
```

În mod normal, eticheta `<A>` fiind o etichetă container, între etichetele de deschidere şi de închidere trebuie să figureze un text. Totuşi, în exemplul de mai sus, dorind să inserăm o ancoră în dreptul titlului, am folosit cea de-a doua construcţie:

```
<A name="ancora1"></A>
<H1 align="center">Ancore definite in acelasi document</H1>
```

Motivul este acela că este considerată o practică incorectă includerea etichetelor de titlu între etichetele `<A>` şi `</A>`.
Se poate utiliza şi construcţia următoare:

```
<H1 align="center"><A name="ancora1">Ancore definite in acelasi
document</A></H1>
```

5. Legătura către o pagină aflată în acelaşi director (folder)

Pentru a realiza o legătură către o pagină aflată în acelaşi director se procedează astfel:

```
<A href="nume_fisier.html">text explicativ</A>
```

unde:

- `href` reprezintă atributul care stabileşte calea către ţinta cu care se face legătura. Dacă fişierul ţintă este în acelaşi director, atributul primeşte ca valoare chiar numele fişierului.
- `text explicativ` - reprezintă textul pe care se face click cu mouse-ul pentru a activa legătura. (De exemplu "Click aici"). Acest text este afişat diferit faţă de restul textului - în general, subliniat şi de culoare albastră.

În Exemplul 6.2 este realizată o legătură reciprocă între două pagini aflate în folderul de lucru *Work*, pagina având aspectul din Figura 6.2.

Exemplul 6. 2

```
<HTML>
  <HEAD>
    <TITLE>legaturi2</TITLE>
  </HEAD>
  <BODY>
    <H1>Pagina 1 </H1><HR>
    <A href="legaturi3.html">Link catre pagina 2 </A>
  </BODY>
</HTML>
```

Salvaţi acest exemplu cu numele *legaturi2.html* iar exemplul următor (Exemplul 6.3) cu numele *legaturi3.html*.

Exemplul 6. 3

```
<HTML>
  <HEAD>
    <TITLE>legaturi3</TITLE>
  </HEAD>
  <BODY>
    <H1>Pagina 2 </H1><HR>
    <A href="legaturi2.html">Link catre pagina 1 </A>
  </BODY>
</HTML>
```

La fel cum aţi procedat şi până acum, deschideţi una dintre cele două pagini cu browserul şi testaţi funcţionarea legăturii dintre ele.
Ambele documente HTML trebuie salvate în acelaşi folder. Veţi observa că atunci când vă aflaţi în Pagina 1 şi faceţi click cu mouse-ul pe textul Link catre Pagina 2 se va deschide cel de-al doilea document HTML şi invers, legătura dintre cele două pagini fiind astfel reciprocă.

Atenţie!

Numele fişierelor care reprezintă valori ale atributului `href` sunt *case sensitive*. Acelaşi lucru se întâmplă şi cu textul care desemnează valorile atributului `name`.

Aceasta înseamnă că fişierul *legaturi5.html* este diferit de fişierul *Legaturi5.html*, iar ancora `<A name="ancora1">` este diferită de `<A name="Ancora1">`
Pentru a evita greşelile, este recomandat să vă denumiţi toate fişierele sau numele pentru ancore cu litere mici.

6. Legătura către o pagină localizată în alt director (folder)

Dacă pagina cu care vrem să facem legătura se află pe acelaşi calculator, dar într-un alt folder, atunci pentru a preciza poziţia ei în structura de directoare se poate folosi adresarea relativă sau adresarea absolută.

Adresarea **absolută** se realizează precizând calea (*path*) completă, pornind de la directorul rădăcină, prin care se poate ajunge la fişierul de care vrem să legăm pagina.
De exemplu, dacă fişierul *culori1.html* se află pe discul *C:,*în folderul *My Documents,* în folderul *Work,* o legătură către el se va face în modul următor:

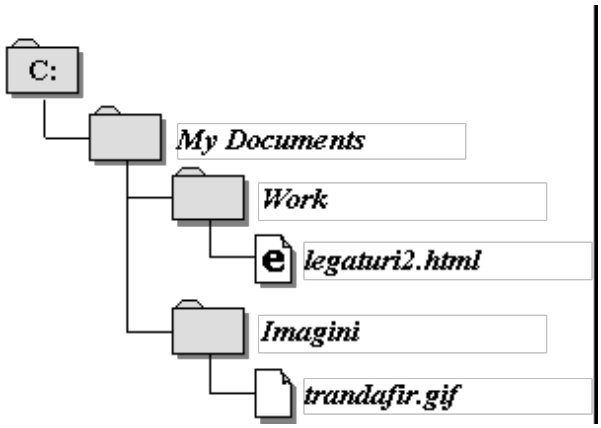
```
<A href="C:/My Documents/Work/culori1.html>Link</A>
```

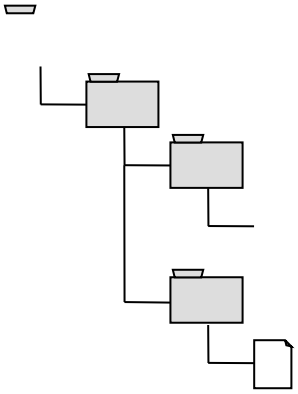
Adresarea **relativă** precizează calea către documentul cu care facem legătura pornind de la documentul în care ne aflăm. Pentru a urca un nivel în structura de directoare se foloseşte şirul de caractere `"../ "`

Exemplu

Să presupunem că ne aflăm într-un document HTML numit *legaturi2.html* plasat în folderul de lucru *Work*. În afară de folderul *Work* în care lucrăm, în *My Documents* se mai află un folder, numit *Imagini*, ca în structura prezentată în Schema 6.1.

Schema 6.1





C:

W
or
k

M
y
D

oc
u
m
e
nt
s

I
m
ag
in
i

le
ga
tu
ri
2.
ht
m
l

tr
a
n
da
fir
.g
if

e

Sc
he
m
a
6.
1

Dorim să realizăm o legătură cu un fişier numit *trandafir.gif* din folderul *Imagini*.
În acest caz, eticheta <A>, plasată în documentul *legaturi2.*html, va avea următoarea formă:

```
<A href="../Imagini/trandafir.gif">Deschide imaginea</A>
```

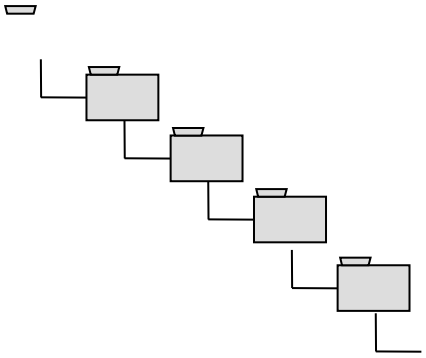
Semnificaţia este următoarea:

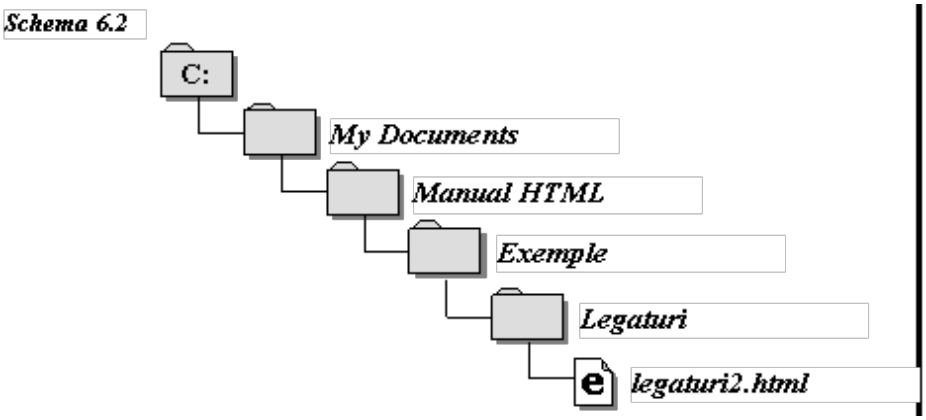
Prin folosirea şirului de caractere "../" (punct punct slash) se urcă un nivel în ierarhia de directoare, în raport cu directorul curent. Prin urmare, întrucât folderul curent, în care se află pagina de pornire este *C:/My Documents/Work,* prin utilizarea şirului de caractere "../" se ajunge în folderul părinte, care este *C:/My Documents.* De aici se continuă calea în

folderul *Imagini,* după care se specifică numele fişierului din acest folder cu care vrem să stabilim legătura.

Exemplu

Să presupunem că ne aflăm în documentul *legaturi2.html* poziţionat ca în Schema 6.2.





Legaturi

C:

Manual HTML

My Documents

Exemple

Schema 6.2

e

Dorim să stabilim o legătură cu un fişier numit *text2.html* care se află în directorul *Exemple* (directorul părinte al folderului nostru, *Legaturi*). Atunci referirea se va face astfel:

```
<A href="../text2.html">Link la text</A>
```

Am urcat un nivel în ierarhie ajungând în directorul *Exemple* şi am specificat numele fişierului cu care dorim să facem legătura.

Dacă fişierul *text2.html* se află cu două nivele mai sus faţă de folderul în care lucrăm, adică dacă se află în folderul *Manual HTML*, atunci adresarea se face astfel:

```
<A href="../../text2.html">Link la text</A>
```

În general, de câte ori este posibil, este de preferat să folosiţi adresarea relativă, pentru ca documentele HTML să fie portabile (mutarea lor să nu invalideze legăturile stabilite între diverse documente).

7. Legătura către pagini externe

O legătură către o pagina externă se realizează simplu, prin utilizarea etichetei `<A> </A>`, specificând adresa URL a paginii ca valoare a atributului `href` astfel:

```
href="http://URL_pagina"
```

Reamintim că specificarea adresei URL se poate face fie folosind numele serverului pe care este stocată pagina fie adresa IP a acestuia. Evident, pentru ca link-ul să funcţioneze, trebuie ca utilizatorul să fie conectat la Internet (lucru valabil pentru toate legăturile externe).

În Exemplul 6.4 este stabilită o legătură către pagina de start Yahoo.

Exemplul 6. 4

```
<HTML>
  <HEAD>
    <TITLE>legaturi4</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Link catre Yahoo.com</H1><HR>
    <A href="http://www.yahoo.com"> Yahoo!</A>
  </BODY>
</HTML>
```

Aspectul paginii este cel din Figura 6.3.

Un atribut util al etichetei `<A>` este **`title`**. Acesta determină apariţia unei mici ferestre (*tool tip*) în pagina Web când mouse-ul se afla pe o legătură, fereastră în care este afişată valoarea dată acestui atribut, care are astfel menirea de a furniza informaţii suplimentare despre semnificaţia unei legături.

Exemplul 6.5 ilustrează utilitatea atributului `title`, aşa cum reiese din Figura 6.4.

Exemplul 6. 5

```
<HTML>
  <HEAD>
    <TITLE>legaturi5</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Atributul title</H1><HR>
    <A href="http://www.google.com" title="Legatura catre
    Google.com">Google</A>
  </BODY>
</HTML>
```

8. Alegerea culorilor pentru legături

Am mai discutat despre acest subiect şi la capitolul despre culori. În mod prestabilit (*default*) se utilizează trei culori pentru legături:

- culoare pentru legăturile **nevizitate** (nu s-a efectuat nici un click pe ele) - albastru
- culoare pentru legăturile **vizitate** (s-a efectuat cel puţin un click pe ele) - violet
- culoare pentru legăturile **active** (deasupra cărora se află mouse-ul la un moment dat, dar încă nu s-a efectuat click) - roşu

Pentru a modifica după preferinţe aceste culori se folosesc cele trei atribute ale etichetei `<BODY>`:

`link` pentru legăturile nevizitate;
`vlink` pentru legăturile vizitate;
`alink` pentru legăturile active.

Valorile pe care le pot lua aceste atribute sunt culori definite prin nume sau conform codului hexazecimal.

Exemplul 6.6 ilustrează modul cum pot fi modificate culorile legăturilor, aşa cum se poate observa din Figura 6.5 care redă aspectul paginii descrise în exemplu.

Exemplul 6. 6

```
<HTML>
  <HEAD>
    <TITLE>legaturi6</TITLE>
  </HEAD>
    <BODY link="yellow" vlink="green" alink="magenta">
    <H1 align="center">Setarea culorilor pentru link-uri</H1><HR>
    <BR>galben pentru legaturi, verde pentru legaturi vizitate si magenta
    pentru legaturi active<BR>
    <A href="legaturi2.html">Link catre pagina 1 </A><BR>
    <A href="legaturi3.html">Link catre pagina 2 </A>
  </BODY>
</HTML>
```

9. Utilizarea poştei electronice (e-mail)

Într-o pagină Web se pot afla legături care permit lansarea în execuţie a aplicaţiei de expediere a mesajelor electronice a celui care vizitează pagina. Făcând click pe textul care însoţeşte legătura, programul de poşta electronică al vizitatorului paginii se va deschide automat, având câmpul adresei

destinatarului, "To:" deja completat cu adresa de mail specificată în pagină. Pentru a realiza acest lucru se foloseşte comanda **`mailto:`** atributul `href` primind o valoare ca mai jos:

```
<A href="mailto:adresa_e-mail">
```

Dacă pagina este vizualizată cu browserul Internet Explorer iar vizitatorul are instalat un e-mail manager, cum sunt de pildă aplicaţiile Microsoft Outlook sau Outlook Expres, activarea legăturii va determina deschiderea unuia dintre aceste programe. În cazul în care managerul de e-mail *default* este de alt tip decât aplicaţia Microsoft, (Eudora, de exemplu) aceasta va porni de asemenea automat, cu o nouă pagină *Send* şi cu adresa destinatarului pre-completată. Dacă pagina este vizualizată în Netscape, se va deschide programul de poştă electronică încorporat în browser.

În exemplul următor (Exemplul 6.7), în momentul când vizitatorul paginii face click pe textul "Trimiteţi un mesaj", aplicaţia de poştă electronică este lansată automat, prin intermediul serviciului `mailto:`, iar câmpul care trebuie completat cu adresa de mail a destinatarului mesajului este de asemenea completat automat cu adresa autor@domeniu.com aşa cum rezultă din Figura 6.6.

Exemplul 6. 7

```
<HTML>
  <HEAD>
    <TITLE>legaturi7</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Expediere de mesaje electronice
    </H1><HR>
    <A href="mailto:autor@domeniu.com" title="adresa mea de mail">
    Trimiteti un mesaj </A>
  </BODY>
</HTML>
```

10. Legături către fişiere oarecare

O pagină Web poate conţine legături nu doar către alte fişiere HTML, dar şi către fişiere de orice alt tip, aflate pe calculatorul propriu sau oriunde pe Web. Ca şi în cazul legăturilor cu alte pagini (documente HTML) vom folosi eticheta `<A> </A>`, astfel:

```
<A href="URL_fisier_destinatie">text explicativ<A>
```

Nefiind vorba despre un fişier HTML, browserul nu va putea să îl proceseze, astfel că va activa procesul de transfer sau de descărcare (*download*), urmând ca, după transferul integral al fişierului, utilizatorul să îl deschidă cu un program adecvat.
În Exemplul 6.8, atunci când se efectuează click pe legătură se deschide caseta de dialog *File download* care permite:

- salvarea fişierului pe disc sau
- deschiderea fişierului în locaţia curentă

Exemplul 6. 8

```
<HTML>
  <HEAD>
    <TITLE>legaturi8</TITLE>
  </HEAD>
```

```
  <BODY>
    <H1 align="center">Legaturi catre fisiere oarecare</H1><HR>
    <A href="html.zip">
    Link catre fisierul download.zip
    </A>
  </BODY>
</HTML>
```

Figura 6.7 redă efectul activării unei legături către un fişier oarecare.

11. Deschiderea paginilor referite printr-o legătură

Pagina nouă, apelată prin activarea unei legături se poate deschide în două moduri, în raport cu pagina sursă:
- în aceeaşi fereastră
- într-o fereastră nouă

În mod prestabilit, legăturile deschid pagina pe care o referă în fereastra curentă. Aceasta înseamnă că dacă veţi face click pe un link, noua pagină se va încărca în locul paginii deja deschise (în aceeaşi instanţă a browserului). Pentru a reveni la pagina anterioară trebuie să apăsaţi butonul *Back* al browserului.

Acest comportament se poate schimba cu ajutorul valorilor atributului **`target`**, asupra căruia vom reveni la capitolul **Cadre**.
Vom menţiona aici doar două dintre aceste valori, corespunzătoare celor două situaţii amintite.
Pentru ca pagina apelată să se deschidă într-o fereastră nouă, se utilizează sintaxa generică de mai jos, în care atributul `target` are valoarea `"_blank"`:

```
<A href="adresa_URL" target="_ blank">text explicativ</A>
```

Pentru ca pagina referită să se deschidă în aceeaşi fereastră cu pagina sursă, atributului i se asociază valoarea `"_self"`:

```
<A href="adresa_URL" target="_self">text explicativ</A>
```

Figura 6.8 redă modul în care se deschide o pagină într-o nouă fereastră, aşa cum este precizat în Exemplul 6.9.

Exemplul 6. 9

```
<HTML>
  <HEAD>
    <TITLE>legaturi9</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Deschiderea paginii in alta fereastra</H1> <HR>
    Pagina de mai jos se va deschide intr-o fereastra noua
    <P>
    <A href="tabel culori.html" target="_blank">Tabelul culorilor</A>
  </BODY>
</HTML>
```

12. Crearea unei bare secundare de navigare

În foarte multe site-uri aţi observat, probabil, existenţa, în partea de jos a paginii, a unui bloc de text care conţine legături către paginile care compun site-ul, legăturile fiind delimitate de mici linii verticale, ca în exemplul de mai jos:

Utilitatea acestei bare este evident legată de facilitarea navigării în site, mai ales în cazurile când dimensiunea paginii depăşeşte un ecran. În acest caz, utilizatorul trebuie să deruleze la citire paginile pe verticală, astfel că link-urile de pe bara de navigare superioară nu mai sunt accesibile.

O astfel de bară secundară de navigare se poate realiza procedând ca în Exemplul 6.10. Evident, legăturile vor face referire la fişierele HTML dorite de dumneavoastră. O pagină care conţine o astfel de bară de navigare va avea aspectul din Figura 6.9.

Exemplul 6. 10

```
<HTML>
  <HEAD>
    <TITLE>legaturi10</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Bara de navigare</H1><HR>
    <H4 align="center">
      <A href="culori1.html">|Culori|</A>
      <A href="fonturi1.html">|Fonturi|</A>
      <A href="text1.html">|Formatarea textului|</A>
    </H4>
  </BODY>
</HTML>
```

Bara verticală inserată între textele care trimit la paginile respective este de obicei plasată pe tastatură pe aceeaşi tastă cu caracterul "\" (*backslash*). Dacă doriţi ca bara să facă parte din textul activ, o veţi insera între etichetele <A> şi </A>, altminteri ea trebuie plasată în afara acestora, astfel:

```
|<A href="culori1.html">Culori</A>|
```

13. Rezumat

Adresa URL constă dintr-un şir de caractere care identifică în mod unic o anumită resursă oferind informaţii despre numele serverului pe care este stocată acea resursă şi despre localizarea ei în structura de directoare de pe server.

Legăturile se introduc într-un document HTML prin intermediul etichetei container <A> </A> conform următoarei sintaxe generale:

```
<A href="adresa_URL" name="nume" title="text" target="tinta">text sau
imagine</A>
```

În funcţie de locul unde este plasat documentul referit există mai multe tipuri de legături:

- Legături în cadrul aceleiaşi pagini (legături interne, ancore). Pentru a crea o legătură internă sunt necesari doi paşi: definirea numelui ancorei şi stabilirea legăturii .

- Legături cu pagini aflate în acelaşi director. Pentru a stabili o legătură cu o pagină aflată în acelaşi director se specifică drept valoare a atributului `href` chiar numele fişierului HTML.

- Legături cu pagini aflate în alt director. Pentru a stabili locaţia documentului cu care facem legătura se poate folosi adresarea relativă (recomandat) sau adresarea absolută.

- Legăturile externe. În cazul legăturilor externe, atributul `href` primeşte ca valoare adresa URL a paginii respective.

Se pot stabili şi legături către fişiere oarecare (nu neapărat documente HTML). La activarea unei astfel de legături se deschide fereastra de *download* a sistemului.

Cu ajutorul comenzii `mailto:` se lansează automat în execuţie aplicaţia de poştă electronică a vizitatorului paginii.

14. Test

41. *Care dintre următoarele afirmaţii este falsă:*

a) Eticheta `<A>` serveşte la stabilirea unei legături în cadrul aceleiaşi pagini
b) Eticheta `<A>` serveşte la stabilirea unei legături către un fişier aflat pe acelaşi calculator
c) Eticheta `<A>` serveşte la scrierea textului cu caractere Arial.

42. *Pentru a preciza numele fişierului spre care se face legătura se foloseşte atributul:*

a) `name`
b) `href`
c) `file`

43. *Care este greşeala din următoarea construcţie?*

```
<H2><A name="#gr">Greseala</A><H2>
```

a) Nu este nici o greşeala.
b) Ar fi trebuit formulat:
   `<H2><A name="gr">Greseala</A></H2>`
c) Ar fi trebuit formulat:
   `<A name="#gr"></A><H2>Greseala</H2>`

44. *Ce realizează exemplul următor?*

```
<A name="sectiunea1">Sectiunea 1</A>
```

a) Inserează o ancoră în pagină şi defineşte numele ei.
b) Stabileşte o legătură în cadrul paginii.
c) Stabileşte o legătura la fişierul *sectiunea1.html*.

45. *Fişierul contact.html, referit în legătura de mai jos:*

```
<A href="../contact.html">Contact</A>
```

este localizat în:

a) acelaşi director ca şi fişierul curent
b) directorul părinte al fişierului curent
c) directorul aflat cu două nivele mai sus decât directorul curent

46. *În documentul culori.html am inclus următoarea ancoră:*

```
<A name="fundal"></A>
```

În acest caz, construcţia corectă pentru a face legătura spre ea din cadrul unui alt document aflat în acelaşi director este:

a) `<A href="#fundal">Background</A>`
b) `<A href="culori.html_fundal">Background</A>`
c) `<A href="culori.html#fundal>Background</A>`

47. *Care este aspectul unei legături într-un text, în mod prestabilit (default)?*

a) de culoare albastră şi subliniată
b) de culoare roşie
c) subliniată

48. *Care dintre următoarele afirmaţii este falsă?*

a) Nu este permisă imbricarea mai multor etichete `<A>`.
b) Nu este permisă schimbarea culorilor legăturilor.
c) Nu este permisă referinţa la alte fişiere decât fişierele HTML.

49. *Pentru a stabili o legătura cu site-ul extern* www.books.com *folosim construcţia:*

a) `<A href="http://www.books.com">`
b) `<A href="file://www.books.com">`
c) `<A href="books.com">`

50. *Fie următoarea etichetă inserată în pagină:*

```
<A href="mailto:utilizator@domeniu.com>Send mail</A>
```

Dacă se execută click pe textul "Send mail":

a) Se deschide pagina www.domeniu.com.
b) Se deschide programul de mail al vizitatorului paginii.
c) Se deschide căsuţa de mail cu adresa specificată.

Legături

1. c)
2. b)
3. b)
4. a)
5. b)
6. c)
7. a)
8. b)
9. a)
10. b)

Capitolul 7
Imagini şi elemente multimedia

Imaginile şi elementele multimedia constituie, fără îndoială, o latură interesantă şi spectaculoasă a oricărei pagini de Web. Puteţi include în paginile dumneavoastră fotografii, imagini animate, sunete sau imagini video. Atunci când sunt folosite în mod judicios, aceste elemente pot îmbogăţi conţinutul paginilor Web, oferindu-le un aspect atractiv şi profesional. Pe de altă parte, folosirea lor în exces poate da paginilor un aspect încărcat şi confuz şi poate conduce la mărirea considerabilă a timpului de încărcare al paginii.

1. Formatele fişierelor grafice

Imaginile sunt stocate în fişiere cu diverse formate, cele mai folosite pe Web fiind, aşa cum era firesc, cele care conferă un raport optim între calitatea imaginii şi dimensiunile fişierului.
Această opţiune este determinată de faptul că majoritatea utilizatorilor Internetului dispun de conexiuni telefonice dial-up, cu viteze destul de mici, astfel că timpul de încărcare al fişierelor, care depinde de dimensiunea acestora, devine un factor determinant.
Două dintre cele mai utilizate tipuri de fişiere grafice sunt **JPEG** (*Joint Photographic Experts Group*) şi **GIF** (*Graphics Interchange Format*).

Formatul GIF

Formatul GIF (*.gif*) foloseşte 256 de culori şi este ideal pentru icon-uri, ilustraţii şi animaţie. Acest format utilizează o tehnologie specială de comprimare care reduce semnificativ dimensiunile fişierelor grafice pentru un transfer mai rapid prin reţea. În procesul de comprimare se păstrează toate caracteristicile imaginii originale, astfel că după decomprimare imaginea are acelaşi aspect ca şi originalul.

Imaginile GIF suportă efecte de transparenţă, întreţesere şi animaţie, asupra cărora vom reveni pe larg în capitolul **Elemente avansate de grafică**.

Deoarece majoritatea browserelor recunosc formatul GIF, acesta a devenit cel mai frecvent utilizat în paginile Web. Poate fi folosit pentru a include imagini direct în pagini (imagini *in-line*) precum şi pentru a referi imaginile prin intermediul unor legături externe.

Totuşi, datorită numărului redus de culori, formatul GIF nu este potrivit pentru fotografii sau imagini de calitate înaltă. Pentru acest tip de imagini, cel mai adecvat este formatul JPEG.

Formatul JPEG

Formatul JPEG (*.jpg*), pe de altă parte, suportă un număr mult mai mare de culori (aproximativ 16 milioane). Dacă doriţi să folosiţi imagini fotografice, formatul JPEG este recomandat, datorită calităţii superioare a imaginii. Dimensiunile unui fişier JPEG nu depind de numărul de culori ci de gradul de comprimare a imaginii. Gradul de comprimare al formatului JPEG este mai ridicat decât cel al formatului GIF. Nu este neobişnuit, de exemplu, ca un fişier GIF care are 200 de Kb să fie comprimat ca fişier JPEG până la dimensiunea de 30 de Kb.

Pentru a realiza un grad atât de înalt de comprimare a imaginilor, formatul JPEG pierde anumite informaţii din imaginea originală. Cu toate că la decomprimare imaginea JPEG nu va fi absolut identică cu imaginea originală, diferenţele vor fi de cele mai multe ori inobservabile.

Formatul JPEG este potrivit pentru imaginile fotografice, dar el nu este adecvat pentru ilustraţii, desene sau imagini de dimensiuni reduse. Algoritmii folosiţi pentru comprimarea şi decomprimarea imaginii alterează în mod notabil zonele de mari dimensiuni colorate cu o singură nuanţă. Din acest motiv, atunci când doriţi să includeţi în pagină un desen sau o ilustraţie care foloseşte un număr redus de culori, formatul GIF este cel mai potrivit.

2. Câteva metode de obţinere a imaginilor

Aveţi la îndemână diverse moduri în care puteţi crea sau procura imagini pe care să le includeţi în paginile dumneavoastră:

- Crearea imaginilor cu ajutorul unui program de grafică - cele mai performante sunt Adobe Photoshop şi CorelDraw.

- Scanarea fotografiilor realizate cu aparate foto tradiţionale şi, eventual, prelucrarea lor ulterioară cu editoare grafice.

- Folosirea aparatelor de fotografiat digitale - deşi sunt nişte echipamente încă destul de costisitoare prezintă marele avantaj că furnizează imagini digitale sub formă de fişiere grafice în formate comune, care se pot descărca direct pe hard-disk, şi care se pot utiliza ca atare sau după o prelucrare grafică minimă.

- Preluarea imaginilor de pe Web. Numărul site-urilor care oferă colecţii de imagini gratuite şi care pot fi utilizate liber este imens. Totuşi, atunci când doriţi să folosiţi în pagina dumneavoastră o imagine care nu face parte dintr-o astfel de colecţie, trebuie să aveţi în vedere aspectul drepturilor de autor. Pentru a prelua o imagine de pe Web trebuie să o salvaţi pe hard-disk-ul dumneavoastră. Puteţi realiza acest lucru astfel:

  - plasaţi cursorul pe imaginea respectivă şi apăsaţi butonul drept al mouse-ului
  - selectaţi din meniul care se deschide *Save Image As...*
  - în fereastra de dialog selectaţi folderul de destinaţie şi numele fişierului
  - apăsaţi butonul *Save*

Este util să vă creaţi unul sau mai multe foldere dedicate imaginilor, organizate pe tipuri (icon, fundaluri, butoane, imagini propriu-zise, etc.), uneori alcătuind adevărate biblioteci, în care să păstraţi toate imaginile pe care intenţionaţi să le folosiţi în paginile dumneavoastră. De asemenea, de mare utilitate sunt aplicaţiile de gestionare a imaginilor, cum este de exemplu cunoscutul ACDSee.

Atenţie!

Dacă imaginea nu este salvată în acelaşi folder în care se află documentul HTML sursă care foloseşte imaginea respectivă, referinţa la imagine trebuie să conţină calea corectă către locaţia ei, altminteri imaginea nu va fi afişată în pagină.

3. Inserarea unei imagini

Pentru a insera o imagine în cadrul unei pagini (o imagine *in-line*), se utilizează eticheta **<IMG>** (de la *image*). Eticheta <IMG> nu este o etichetă container, prin urmare nu necesită o etichetă corespunzătoare de închidere.

Pentru a putea identifica imaginea care va fi inserată, se utilizează atributul **src** (*source*) al etichetei <IMG>. Atributul src îi comunica browserului numele şi locaţia imaginii care urmează să fie inserată. Valoarea acestui atribut este adresa URL a imaginii respective.

Dacă imaginea se află în acelaşi director cu fişierul HTML care face referire la imagine, atunci adresa URL a imaginii este formata numai din numele fişierului, inclusiv extensia.

```
<IMG src="imagine.extensie">
```

Dacă imaginea se afla într-un alt director, URL-ul imaginii trebuie specificat fie prin adresarea absolută fie, preferabil, prin cea relativă.
Exemplul 7.1 ilustrează modul în care se pot insera într-o pagină două imagini, una dintre ele aflându-se în acelaşi folder ca şi pagina sursă, iar cealaltă în folderul *Imagini*. Figura 7.1 redă aspectul paginii descrise în exemplu.

Exemplul 7. 1

```
<HTML>
  <HEAD>
    <TITLE>imagini1</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Imagini in pagina</H1><HR>
    Imagine aflata in acelasi folder<P>
    <IMG src="tiger.gif" border="5"><P>
    Imagine aflata in folderul Imagini<P>
    <IMG src="../Imagini/bernese.jpg" border="1" >
  </BODY>
</HTML>
```

În acest exemplu este prezent şi atributul **border** al etichetei <IMG>. Acesta este folosit pentru a plasa un chenar în jurul imaginii. Valoarea atributului border este numărul de pixeli care reprezintă grosimea chenarului din jurul imaginii. Absenţa atributului sau setarea la valoarea "0" face ca acest chenar să nu fie prezent.

Un alt atribut util asociat etichetei <IMG> este **alt**. Acest atribut permite afişarea unui text explicativ în spaţiul în care va fi afişată imaginea în pagină.

În Exemplul 7.2 este ilustrată utilitatea atributului alt.

Exemplul 7. 2

```
<HTML>
  <HEAD>
    <TITLE>imagini2</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center"> Atributul alt </H1><HR>
    Acesta este un...<BR>
    <IMG src="../Imagini/orangerose" alt="trandafir">
  </BODY>
</HTML>
```

După cum puteţi observa din Figura 7.2 imaginea pe care am inclus-o în document nu este afişată. Motivul este că am omis intenţionat extensia *.gif* a fişierului pentru a exemplifica utilitatea atributului alt. Prin urmare, dacă dintr-

un motiv oarecare, imaginea nu se încarcă în pagină, se încarcă mai greu, sau este vizualizată cu un browser care nu suportă grafica, în zona rezervată imaginii va fi afişat textul specificat ca valoare a atributului `alt`.
De asemenea, textul specificat ca valoare pentru atributul `alt` va fi afişat şi în cadrul unei mici ferestre care se deschide în momentul când cursorul mouse-ului este menţinut deasupra imaginii.

Figura 7.3 redă aspectul paginii în cazul când atributul `src` are valoarea corectă: `"orangerose.gif"`.

Un alt motiv pentru care este indicată folosirea atributului `alt` este acela că vizitatorul paginii are posibilitatea de a vedea încă înainte de încărcarea completă a imaginii ce anume se va afişa în zona respectivă. Astfel, el are posibilitatea de a aştepta încărcarea completă a imaginii sau de a trece la altă pagină.

4. Dimensionarea imaginii

Dimensionarea imaginii se realizează cu ajutorul atributelor **`width`** - prin care se stabileşte lăţimea imaginii şi **`height`** - prin care se stabileşte înălţimea imaginii.

În Exemplul 7.3 sunt precizate şi dimensiunile imaginii incluse în document. Se observă că dimensionarea imaginilor se face în pixeli. Este, desigur, posibil, ca dimensionarea să se facă în procente, ca şi la liniile orizontale, de exemplu. Totuşi, în afara unor cazuri speciale, aceasta este considerată o practică greşită, deoarece browserul va redimensiona imaginea iar rezultatele vor fi de calitate slabă.

Exemplul 7. 3

```
<HTML>
  <HEAD>
    <TITLE>imagini3</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Dimensionarea imaginilor</H1><HR>
    <IMG src="tiger.gif" width="350" height="250"><P>
  </BODY>
</HTML>
```

Observând Figura 7.4 care redă aspectul paginii descrise mai sus putem remarca faptul că setarea dimensiunilor a afectat calitatea imaginii deoarece dimensiunile originale ale imaginii erau mai mici. Dimensiunile originale ale imaginii *tiger.gif*, determinate cu ajutorul unui editor grafic, sunt: lăţime=200 pixeli, înălţime=125 pixeli.

Dacă dimensiunile imaginii în pagină sunt setate la valori considerabil mai mari decât dimensiunile imaginii originale calitatea imaginii incluse în pagină va avea de suferit. Concluzia care se impune este că dimensionarea imaginilor din pagină trebuie să se facă la valori cât mai apropiate de dimensiunile iniţiale ale imaginilor.

Vă puteţi întreba atunci, de ce mai este necesară includerea atributelor de dimensionare a imaginii. Nu numai că este necesară, dar este considerată o practică defectuoasă absenţa lor din cadrul etichetei `<IMG>`.

Motivul este acela că includerea dimensiunilor imaginii oferă browserului posibilitatea de a rezerva spaţiu pentru imagine şi de a începe încărcarea textului simultan cu încărcarea imaginii. Dacă atributele de dimensionare nu

sunt prezente, browserul va efectua nişte paşi suplimentari pentru a calcula spaţiul din pagină necesar afişării imaginii. Din acest motiv afişarea textului nu va putea începe decât după ce imaginea este încărcată în întregime.

Este indicat să evitam o asemenea situaţie deoarece imaginile se încarcă mai greu decât textul şi mulţi vizitatori ai paginii nu vor aştepta suficient pentru ca întreaga pagină (imagini şi text) să fie încărcată.

5. Alinierea imaginii şi a textului

Alinierea unei imagini în raport cu textul din pagină se realizează prin intermediul atributului **`align`**, care poate lua următoarele valori:

- `left` - aliniere la stânga; textul este dispus în partea dreaptă a imaginii încadrând imaginea
- `right` - aliniere la dreapta; textul este dispus în partea stânga a imaginii încadrând imaginea
- `top` - aliniere deasupra; partea de sus a imaginii se aliniază cu prima linie a textului ce precede imaginea
- `middle` - aliniere la mijloc; mijlocul imaginii se aliniază cu prima linie a textului ce precede imaginea.
- `bottom` - aliniere dedesubt, la bază; partea de jos a imaginii se aliniază cu prima linie a textului.

Valorile `left`, `right`, şi `bottom` ale atributului `align` permit ca textul să fie dispus în jurul imaginii, în vreme ce `top` şi `middle` nu permit acest lucru.

Exemplul 7.4 ilustrează modul de aliniere `bottom`, aşa cum rezultă din Figura 7.5.

Exemplul 7. 4

```
<HTML>
  <HEAD>
    <TITLE>imagini4</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Alinierea imaginii si textului (bottom)</H1><HR>
    <IMG src="../Imagini/eaglehed.gif" align="bottom" width="100"
    height="66" alt="vultur">
    Imagine text.Imagine text.Imagine text.Imagine text.Imagine
    text.Imagine text.Imagine text.Imagine text.Imagine text.Imagine
    text.Imagine text.Imagine text.Imagine text.Imagine text.Imagine
    text.Imagine text.
    Imagine text.Imagine text.Imagine text.Imagine text.Imagine
    text.Imagine text.Imagine text.Imagine text.
  </BODY>
</HTML>
```

Figura 7.6 redă modul de aliniere `left` iar Figura 7.7 modul de aliniere `right`. Din Figura 7.8 şi Figura 7.9 puteţi observa modurile de aliniere `top` şi `middle` precum şi faptul că aceste alinieri nu permit dispunerea textului în jurul imaginii.

Alte două atribute utile, care servesc la alinierea imaginii faţă de restul elementelor din pagină, sunt atributele **`hspace`** şi **`vspace`**. Ele precizează distanţa, în pixeli, pe orizontală, respectiv pe verticală, dintre imagine şi restul elementelor din pagină.
În Exemplul 7. 4 vom schimba modul de aliniere în cadrul etichetei `<IMG>` şi vom adăuga atributele `hspace`, respectiv `vspace`, astfel:

```
<IMG src="../Imagini/eaglehed.gif" align="left" width="100" height="66"
alt="vultur" vspace="10" hspace="10">
```

Aspectul paginii va fi cel din Figura 7.10.

Aţi observat, probabil, că dintre valorile pe care le poate lua atributul `align` lipseşte valoarea `center`. Într-adevăr, alinierea unei imagini la centrul paginii nu se poate face prin intermediul atributului `align`. Centrarea unei imagini se poate realiza numai dacă este izolată de textul care o înconjoară. Pentru aceasta se poate folosi eticheta `<CENTER>` sau se poate include imaginea într-un bloc paragraf sau într-un bloc `<DIV>` având atributul `align` setat la valoarea `center`.

Exemplul 7.5 ilustrează acest mod de aliniere folosind eticheta `<DIV>` aspectul paginii fiind cel din Figura 7.11.

Exemplul 7. 5

```
<HTML>
  <HEAD>
    <TITLE>imagini5</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Centrarea unui imagini</H1><HR>
    Imagine text.Imagine text.Imagine text.Imagine text.Imagine
    text.Imagine text.Imagine text.Imagine text.
    <DIV align="center">
      <IMG src="../Imagini/devil.gif" width="64" height="64"
      alt="dracusor">
    </DIV>
    Imagine text.Imagine text.Imagine text.Imagine text.Imagine
    text.Imagine text.Imagine text.Imagine text.
  </BODY>
</HTML>
```

Exemplul 7.6 ilustrează modul în care pot fi aliniate două imagini faţă de textul din pagină.
Din Figura 7.12 puteţi observa că, dacă alinierea la stânga a imaginii împreună cu folosirea atributelor `hspace` şi `vspace` conduce la un aspect ordonat al elementelor, aspectul textului în raport cu imaginea aliniată la dreapta depinde de dimensiunea ferestrei browserului.

Exemplul 7. 6

```
<HTML>
  <HEAD>
    <TITLE>imagini6</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Alinierea a doua imagini</H1><HR>
    <IMG src="../Imagini/mtnscene.jpg" align="left" width="200"
    height="144" align="left" hspace="10" vspace="10">
    Imagine text. Imagine text. Imagine text. Imagine text.Imagine text.
    Imagine text. Imagine text. Imagine text.
    <IMG src="../Imagini/sunscene.jpg" align="right" width="202"
    height="132" hspace="10" vspace="10">
    Imagine text. Imagine text. Imagine text. Imagine text.Imagine text.
    Imagine text. Imagine text. Imagine text.Imagine text. Imagine text.
    Imagine text. Imagine text.Imagine text. Imagine text. Imagine text.
    Imagine text.Imagine text. Imagine text. Imagine text. Imagine
    text.Imagine text. Imagine text. Imagine text. Imagine text.
```

```
  </BODY>
</HTML>
```

6. Imagini folosite ca fond (background) al paginii

O imagine poate fi utilizată şi pentru a stabili fondul unei pagini Web. În acest scop se foloseşte atributul **background** al etichetei <BODY>, având ca valoare adresa URL a imaginii. Imaginea se multiplică aliniat (*tiling*) pe orizontală şi pe verticală până umple întregul ecran.
Exemplul 7.7 ilustrează utilizarea atributului background.

Exemplul 7. 7

```
<HTML>
  <HEAD>
    <TITLE>imagini7</TITLE>
  </HEAD>
    <BODY background="../Imagini/silk.jpg">
      <H1 align="center">Imaginea ca fond al paginii</H1><HR>
      Fond de matase...
    </BODY>
</HTML>
```

Figura 7.13 redă aspectul unei pagini care foloseşte ca fundal o imagine.

7. Imagini folosite ca legături

Pentru a folosi o imagine drept legătura se procedează ca în următorul exemplu:

```
<A href="tabel_culori.html">
  <IMG src="prism.gif" width="100" height="80" alt="culori">
</A>
```

S-a folosit eticheta <A> prin intermediul căreia am creat legătura cu fişierul tabel_culori.html. Între etichetele <A> şi </A> am inclus o imagine care înlocuieşte textul explicativ pe care vizitatorul urmează să facă click cu mouse-ul. Imaginea serveşte, deci, ca legătură către o anumită pagină. Imaginea folosită ca legătura este *prism.gif* iar pagina spre care este făcută legătura este *tabel_culori.html*. În mod prestabilit imaginea utilizată pe post de zonă activă este înconjurată de un chenar având culoarea unei legături (albastru). Dacă stabilim pentru atributul border al etichetei valoarea "0" acest chenar dispare.
Exemplul 7.8 prezintă codul complet.

Exemplul 7. 8

```
<HTML>
  <HEAD>
    <TITLE>imagini8</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center"> Imagini cu legaturi </H1><HR>
    <BR>Am pus o imagine cu legatura pe pagina<P>
    <A href="tabel culori.html">
    <IMG src="../Imagini/prism.gif" width="100" height="80" alt="culori">
    </A>
  </BODY>
</HTML>
```

În Figura 7.14 puteţi observa modul în care funcţionează imaginea folosită ca legătură în exemplul de mai sus.

8. Imaginile miniaturale (thumbnails)

Dacă într-o pagină Web este necesară afişarea unui număr mare de imagini, încărcarea paginii se va face destul de greu. De multe ori, vizitatorii paginii nu vor avea răbdarea necesară pentru a aştepta încărcarea integrală a paginii.

O soluţie de compromis frecvent aplicată pentru scurtarea timpului de încărcare a unei pagini ce prezintă multe imagini o reprezintă utilizarea imaginilor miniaturale (*thumbnail*-uri). O imagine *thumbnail* este o versiune de dimensiuni reduse a unei imagini mai mari, care se încarcă mult mai rapid şi care, deşi este de dimensiuni mici şi de calitate scăzută, permite vizitatorului să afle ce anume reprezintă şi să decidă dacă este interesat sau nu să deschidă versiunea integrală..
Imaginea miniaturală reprezintă o legătură spre fişierul cu imaginea originală din care provine. În cazul în care vizitatorul paginii doreşte să vadă imaginea originală, o poate deschide efectuând click imaginea *thumbnail.*

Realizarea imaginilor miniaturale se poate face fie prin micşorarea proporţională a imaginilor, fie prin reducerea lor la o dimensiune prestabilită (*strech*) chiar dacă imaginea se distorsionează, în cazul în care se doreşte, de pildă, ca toate *thumbnail*-urile dintr-o pagină să aibă aceleaşi dimensiuni.

Există două metode pentru a plasa o imagine *thumbnail* în pagină:

- folosind două imagini, una fiind imaginea *thumbnail* iar cealaltă, imaginea originală
- folosind o singură imagine, redimensionată (micşorată) direct în pagină

Folosirea a două imagini

O imagine *thumbnail* este foarte uşor de realizat. Oricare ar fi programul de grafică pe care îl folosiţi, tot ceea ce aveţi de făcut este să găsiţi opţiunea *Resize* (care de obicei face parte din meniul *Edit* sau *Image*) şi să precizaţi în caseta de dialog care se deschide, dimensiunile dorite. Veţi observa că există de regulă o opţiune setabilă pentru păstrarea proporţionalităţii cu dimensiunile. Dacă este activată, după stabilirea uneia dintre dimensiuni (*width*, de exemplu), cealaltă (*height*) este calculată automat.

În Exemplul 7.9 este plasată o imagine *thumbnail* în pagină, folosind prima metodă, Figura 7.15 ilustrând funcţionarea imaginii *thumbnail*.

Exemplul 7. 9

```
<HTML>
  <HEAD>
    <TITLE>imagini9</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Thumbnails - doua imagini</H1><HR>
    <P>
    <A href="../Imagini/zdog.gif">
    <IMG src="../Imagini/zdog_th.gif"></A>
  </BODY>
<HTML>
```

Fişierul *zdog_th.gif* conţine imaginea *thumbnail* care este folosită ca legătură, pentru a referi imaginea originală, *zdog.gif*. Deoarece imaginea *thumbnail* este

folosită ca o legătura către imaginea originală, în jurul ei este afişat un chenar prestabilit albastru. Dacă doriţi ca acest chenar să nu mai fie afişat, puteţi folosi atributul `border` al etichetei `<IMG>` setat la valoarea "0". Imaginea originală se va deschide în aceeaşi fereastră cu pagina în care este plasată imaginea *thumbnail*. Pentru a reveni în pagină, trebuie să apăsaţi butonul *Back* al browserului.

Folosirea unei singure imagini

Această metodă este adesea preferată, deoarece foloseşte un singur fişier, cel care conţine imaginea originală, miniaturizarea ei făcându-se chiar în pagină, cu ajutorul atributelor `width` şi `height`. Este o metoda mai eficientă şi mai rapidă, deoarece imaginea de dimensiuni mari care urmează a fi afişată este deja încărcată în memoria *cache* a browserului şi este imediat disponibilă pentru afişare.

În exemplul următor (Exemplul 7.10) am folosit o singură imagine pentru a încărca atât imaginea *thumbnail* cât şi pe cea cu dimensiunile originale. Aspectul paginii este redat în Figura 7.16.

Exemplul 7. 10

```
<HTML>
  <HEAD>
    <TITLE>imagini10</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Thumbnails - o singura imagine</H1><HR>
    <P>
    <A href="../Imagini/zdog.gif">
    <IMG src="../Imagini/zdog.gif" width="70" height="76"></A>
  </BODY>
<HTML>
```

După cum se poate observa, am creat un link (cu ajutorul etichetei `<A>`) la fişierul *zdog.gif* care conţine imaginea originală. Am folosit drept legătură aceeaşi imagine, dar redimensionată, folosind atributele `width` şi `height`.

În exemplul anterior, imaginea *thumbnail* este încărcată prima. În momentul când vizitatorul face *click* pe ea, este încărcată imaginea cu dimensiunile originale. Avantajul ultimei metode este faptul că imaginea este încărcată de la început şi redimensionată pentru *thumbnail*, ceea ce face ca afişarea imaginii cu dimensiunile originale să fie mai rapidă.

9. Imaginile video

Pentru a insera o imagine video într-un document HTML se folosesc atributele **`dynsrc`**, **`controls`**, **`loop`** şi **`start`** ale etichetei `<IMG>`.
Atributul **`dynsrc`** înlocuieşte atributul `src` şi permite inserarea în documentul HTML a unei imagini video în acelaşi mod în care este inserată o imagine statică.
Valoarea atributului `dynsrc` este adresa URL a fişierului video care va fi inclus în pagină conform sintaxei:

```
<IMG dynsrc="URL_fisier_video">
```

Acest atribut este o extensie Internet Explorer şi nu este recunoscut de browserele Netscape. Pentru a vizualiza o imagine video *in-line* într-un browser Netscape, vizitatorul paginii trebuie să instaleze un program auxiliar de tip

plug-in. Dacă programul *plug-in* nu este disponibil pe computerul vizitatorului, imaginea video nu va putea fi afişată.

Singurul format de fişiere video care este suportat de extensiile Internet Explorer este **AVI** (*Audio Video Interleave*), deoarece acesta este formatul de redare care este inclus în browser. Construcţia de mai jos include într-un document HTML fişierul video *introducere.avi* care se află în folderul *Video*:

```
<IMG dynsrc="../Video/introducere.avi">
```

Efectul acestei etichete este deschiderea de către browser a unei ferestre de vizualizare în interiorul paginii Web, fereastră în care va rula clipul video *introducere.avi*, inclusiv sunetul, dacă acesta face parte din clip şi computerul vizitatorului este setat să redea sunetele. Ca şi imaginile obişnuite, imaginea video este afişată pe măsură ce este încărcată.

Deoarece nici un alt browser în afară de Internet Explorer nu recunoaşte această extensie, este recomandat să includeţi în cadrul etichetei `<IMG>` şi atributul `src` prin care să furnizaţi o imagine statică ce va fi afişată în acelaşi cadru. Browserele care nu recunosc extensia `dynsrc` vor afişa imaginea statică pe când Internet Explorer va afişa imaginea video. Ordinea în care apar cele două atribute nu are importanţă. De exemplu:

```
<IMG dynsrc="../Video/intro.avi" src="../Imagini/intro.gif">
```

În mod normal, Internet Explorer redă clipul video într-o fereastră în care nu sunt afişate nici un fel de butoane de control. Utilizatorul poate relua, opri sau continua redarea clipului efectuând click dreapta cu mouse-ul în interiorul ferestrei.
Pentru a adăuga butoane de control acestei ferestre se utilizează atributul **`controls`** al etichetei `<IMG>`. Atributul `controls` nu are alocată nici o valoare, prezenţa sa având doar scopul de a adăuga butoanele de control asemănătoare celor de la aparatele video. De exemplu:

```
<IMG dynsrc="../Video/intro.avi" src="../Imagini/intro.gif" controls>
```

Clipul video inclus în pagină este redat de browser o singură dată, de la început până la sfârşit. Pentru a repeta redarea clipului de un anumit număr de ori este folosit atributul **`loop`**. Valorile posibile ale atributului sunt:

- un număr întreg care reprezintă numărul de reluări ale clipului
- `infinite`, caz în care clipul este redat până când utilizatorul stopează derularea sa apăsând butonul stop al ferestrei de vizualizare (în cazul când fereastra conţine butoanele de control) sau efectuează click dreapta cu mouse-ul în fereastra de vizualizare.

Exemplul de mai jos ilustrează modul în care poate fi setat atributul `loop`:

```
<IMG dynsrc="../Video/intro.avi" src="../Imagini/intro.gif" controls
loop="infinite">
```

Aşa cum am precizat, redarea imaginii video începe imediat ce aceasta a fost complet încărcată în pagină. Pentru a schimba acest comportament se foloseşte atributul **`start`** care poate avea valorile:

- `mouseover`, situaţie în care derularea imaginii video începe în momentul când mouse-ul este plasat deasupra imaginii
- `fileopen`, valoarea predefinită, situaţie în care derularea imaginii începe imediat după încărcarea în pagină

Cele două valori pot fi combinate pentru a se realiza redarea imaginii mai întâi imediat după încărcarea în pagină şi apoi de fiecare dată când mouse-ul este plasat deasupra ei, ca în exemplul următor:

```
<IMG dynsrc="../Video/intro.avi" src="../Imagini/intro.gif" controls
loop="infinite" start="fileopen, mouseover">
```

Imaginile video *in-line* pot fi tratate ca şi imaginile statice. O astfel de imagine poate fi aliniată folosind atributul **align**, sau poate fi spaţiată faţă de textul care o înconjoară.
Exemplul 7.11 ilustrează atributele prezentate mai sus. Pagina va avea aspectul din Figura 7.17.

Exemplul 7. 11

```
<HTML>
  <HEAD>
    <TITLE>imagini11</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Imagini video</H1><HR>
    <FONT size="4" color="blue">Clipul video de mai jos face parte din
    jocul Heroes 3</FONT><P>
    <CENTER>
    <IMG dynsrc="../Video/shield.avi" src="../Imagini/banana.gif" controls
    start="fileopen, mouseover" loop="infinite">
    </CENTER>
  </BODY>
</HTML>
```

10. Sunetele

Dacă eticheta <IMG> permite afişarea unei imagini de fundal, există şi o etichetă care realizează includerea în pagina Web a unei muzici de fundal şi anume eticheta **<BGSOUND>**. Această etichetă este, de asemenea, o extensie Internet Explorer deci nu este recunoscută şi executată în alte browsere.

Browserul Internet Explorer conţine un decodor de sunet încorporat şi permite integrarea prin eticheta <BGSOUND> a sunetului de fundal pentru o pagină.
Sintaxa etichetei <BGSOUND> este următoarea:

```
<BGSOUND src="URL_fisier_sunet" loop="valoare">
```

Atributul **src** are drept valoare adresa URL a fişierului de sunet care este folosit ca fundal sonor al paginii.
În mod curent, Internet Explorer recunoaşte trei tipuri de fişiere de sunet:

- fişiere cu extensia ***.wav*** care este formatul nativ pentru PC
- fişiere cu extensia ***.au***, formatul nativ pentru sistemele UNIX
- fişiere cu extensia ***.midi***, un format universal acceptat pentru codificarea sunetelor

Pentru a include într-o pagină un fişier de sunet se procedează ca în exemplul următor:

```
<BGSOUND src="../Sunet/wellcome.wav>
```

Ca şi în cazul imaginilor video *in-line*, fişierul de sunet este redat o singură dată, la încărcarea paginii. Pentru redarea sa repetată se foloseşte atributul **loop** al etichetei <BGSOUND> care poate avea ca valori:

- un număr întreg, care reprezintă numărul de reluări ale piesei
- `infinite`, caz în care piesa muzicală este reluată până când utilizatorul părăseşte pagina sau închide fereastra browserului.

Executaţi Exemplul 7.12 cu un browser Internet Explorer pentru a observa efectul etichetei `<BGSOUND>`. Desigur, este necesar să schimbaţi adresa URL a fişierului de sunet specificând un fişier existent pe hard-disk-ul dumneavoastră.

Exemplul 7. 12

```
<HTML>
  <HEAD>
    <TITLE>imagini12</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Muzica de fundal</H1><HR>
    <P>
    <FONT size="4" color="red">Muzica se va auzi pana cand veti inchide
    pagina</FONT>
    <BGSOUND src="../Sunet/Welcom98.wav" loop="infinite">
  </BODY>
</HTML>
```

11. Sugestii privind folosirea imaginilor şi elementelor multimedia

Când folosiţi imagini într-o pagină Web, trebuie să vă puneţi întrebarea: "Este necesară această imagine sau ar fi suficient să folosesc text?". Chiar dacă se spune că o imagine face cât o mie de cuvinte, acest lucru nu este întotdeauna adevărat pe Web. Înlocuirea textului care conţine informaţiile esenţiale dintr-o pagină Web cu imagini este adesea o greşeală.
Un prim motiv este că încă mai există browsere non-grafice cum este Lynx, care nu afişează decât textul.
Apoi, chiar şi în browserele grafice, anumiţi utilizatori ar putea opta, din varii motive, pentru dezactivarea afişării imaginilor.
Un alt motiv ar fi acela că, din cauza duratei mari de încărcare a fişierelor cu imagini de mari dimensiuni, unii utilizatori ar putea renunţa pur şi simplu la deschiderea paginii.
Şi nu în ultimul rând, deoarece, spre deosebire de text, imaginile pot întâmpina diverse dificultăţi tehnice la afişare.

Imaginile dintr-o pagină Web trebuie gândite ca nişte instrumente vizuale care au rolul de a susţine şi ilustra conţinutul scris al paginii. De exemplu, într-un catalog *on-line* imaginile produselor prezentate sunt esenţiale pentru conţinutul paginii respective. Icon-urile cu rol de legături sau instrumente de navigaţie pot reprezenta repere vizuale deosebit de utile în cadrul unui site. Dar dacă o imagine nu serveşte unui scop bine definit în pagină, e bine să vă gândiţi de două ori, înainte de a o folosi.

De asemenea se recomandă să fiţi precaut şi în privinţa imaginilor folosite ca fundal (*background*).
Într-adevăr, acestea pot oferi un aspect spectaculos paginii dumneavoastră. Însă trebuie să fiţi conştient de faptul că o imagine folosită ca fundal va creşte cu siguranţă timpul necesar încărcării paginii. Dacă în cazul imaginilor *in-line* textul se poate afişa înaintea sau în timpul încărcării imaginii, la imaginile de fundal afişarea celorlalte elemente din pagină nu va putea începe decât după încărcarea în întregime a fundalului.

Mai mult, spre deosebire de imaginile *in-line,* care pot îmbogăţi conţinutul informaţional al paginilor, imaginile de fundal nu aduc cu adevărat decât foarte

rar un supliment de informaţie. Deşi aspectul unei asemenea pagini poate fi atrăgător, de cele mai multe ori folosirea imaginilor de fundal în combinaţii nefericite cu fonturi de anumite tipuri sau culori, fără contrast suficient, poate face pagina încărcată şi greu de citit.

Dacă totuşi vă decideţi să folosiţi o imagine ca fundal al paginii este recomandat să specificaţi simultan şi o culoare de fundal, adică să setaţi atributul `bgcolor`. Astfel, dacă dintr-un motiv oarecare imaginea de fundal nu se încarcă, pagina va avea totuşi culoarea prestabilită de dumneavoastră.

Ca şi imaginile, şi celelalte elemente multimedia trebuie folosite cu discernământ într-o pagină Web.
Imaginile video au dimensiuni semnificativ mai ari decât imaginile statice, deci se încarcă mult mai încet decât acestea astfel încât includerea lor *in-line* va mări considerabil timpul de încărcare al paginii.
În plus, formatele diverse utilizate nu pot fi deschise întotdeauna cu browserul, unele impunând existenţa unor *plug-in*-uri sau chiar unor programe speciale.

O metodă de compromis recomandată este descrierea conţinutului fişierului, eventual inserarea unui fragment video exemplificativ (*sample*) de dimensiuni mult mai mici, similar cumva metodei *thumbnail*-urilor. Se asigură o legătură şi spre fişierul integral, astfel încât, după vizualizarea exemplului, vizitatorul să poată alege dacă va aştepta timpul necesar încărcării sale sau nu.

Muzica de fundal este de cele mai multe ori nerecomandată. Foarte mulţi vizitatori pot vizualiza paginile dumneavoastră având sunetul dezactivat. Folosirea unei muzici de fundal încetineşte încărcarea paginii şi, în plus, este foarte posibil ca muzica pe care aţi ales-o să nu fie pe gustul celor care vă vizitează pagina, ba chiar să îi determine să o părăsească înainte de a citi conţinutul acesteia.

Dacă este necesar, totuşi, să introduceţi muzică sau sunet în paginile dumneavoastră, este o prevedere înţeleaptă să plasaţi fişierele de sunet separat şi să asiguraţi legături către acestea astfel încât vizitatorul să le poată audia, dacă doreşte.

12. Imaginile şi timpul de încărcare al paginii

Unul dintre cele mai importante aspecte care trebuie luate în considerare la includerea imaginilor într-un document este **timpul de încărcare** a documentului.

Durata de încărcare a paginilor depinde de mulţi factori. Depinde de modemul şi conexiunea vizitatorului, de performanţele serverului gazdă, de trafic, de lăţimea de bandă, etc. Cum nu puteţi avea control asupra acestora, singurul lucru care vă rămâne de făcut este să vă construiţi paginile astfel încât să aibă o bună viteză de încărcare chiar şi în cazul unui vizitator care deţine o conexiune modestă ca performanţe.

Pe lângă alegerea cu foarte multă grijă a imaginilor care vor fi incluse în document, mai există câteva modalităţi prin care poate fi ameliorată durata de încărcare a paginii:

- Optimizarea imaginilor. Folosiţi-vă în mod judicios de instrumentele de control al imaginilor puse la dispoziţie de editorul grafic folosit, optimizaţi dimensiunile imaginii şi numărul de culori la cât mai puţine posibil. Încercaţi să găsiţi un

raport optim între dimensiunea fişierului şi calitatea imaginii. Evitaţi fotografiile sau imaginile de fundal de dimensiuni foarte mari.

- Reutilizarea imaginilor. Această metodă este eficientă mai ales în cazul icon-urilor sau elementelor grafice de navigaţie care sunt prezente în mai multe pagini pe parcursul unui site. Cele mai multe browsere reţin în memoria *cache* elementele documentelor care urmează a fi afişate. Astfel, dacă o imagine este utilizată în mai multe pagini aceasta nu trebuie încărcată de fiecare dată în memorie ci este disponibilă pentru a fi afişată oricând se face referirea la ea.

- Divizarea documentelor de dimensiuni mari în mai multe documente de dimensiuni reduse. Această regulă generală include şi paginile care conţin imagini *in-line*. Mai multe documente de dimensiuni mai mici legate între ele prin legături sunt mai bine acceptate de vizitatori decât un singur document foarte mare care necesită un timp de încărcare îndelungat. Regula general acceptată este menţinerea dimensiunilor unui document Web în jurul valorii de 50Kb, incluzând aici şi imaginile, desigur.

- Folosirea imaginilor *thumbnail*. Dacă pagina conţine un mare număr de imagini, folosiţi imagini miniaturale care să refere imaginile originale. În plus, întrucât imaginea originală referită prin imaginea *thumbnail* se poate deschide într-o nouă fereastră, nefiind asociată cu restul elementelor din pagină, este mai comod pentru vizitator să o salveze pe computerul propriu pentru o vizualizare ulterioară.

- Specificarea dimensiunile imaginilor. În acest mod este evitată etapa calculării de către browser a spaţiului necesar pentru afişarea imaginii, îmbunătăţindu-se viteza de încărcare a paginii.

13. Rezumat

Formatele cele mai utilizate pentru fişierele grafice sunt JPEG şi GIF.
Inserarea unei imagini într-un document HTML se realizează cu ajutorul etichetei `<IMG>`.
Sintaxa etichetei `<IMG>` este următoarea:

```
<IMG src="URL_imagine" align="pozitie" width="valoare" height="valoare"
vspace="valoare" hspace="valoare" alt="text" border="valoare"
dynsrc="URL_imagine_video" loop="valoare" start="valoare" controls>
```

unde:
- `src` precizează locaţia imaginii
- `width` şi `height` stabilesc dimensiunile imaginii
- `vspace` şi `hspace` stabilesc distanţa pe verticală şi orizontală faţă de restul textului din pagină
- `align` aliniază imaginea faţă de textul din jurul ei
- `alt` furnizează un text explicativ asupra imaginii
- `border` stabileşte grosimea chenarului imaginii
- `dynsrc` introduce o imagine video şi precizează locaţia ei
- `loop` specifică numărul de reluări ale imaginii video
- `start` precizează momentul începerii redării imaginii
- `controls` afişează butoanele de control ale ferestrei video

Imaginile pot fi folosite ca fond al paginii cu ajutorul atributului `background` al etichetei `<BODY>`.

O imagine poate servi drept legătură către un alt document HTML prin includerea etichetei `<IMG>` între etichetele `<A> </A>`.

O practică des folosită este inserarea în pagină a imaginilor miniaturale care la executarea unui *click* pe ele deschid imaginea originală.

Muzica de fundal poate fi adăugată unui document prin intermediul etichetei `<BGSOUND>`.

Alegerea imaginilor care vor fi incluse într-un document HTML trebuie făcută cu grijă, în funcţie de rolul acestora în structura documentului şi având permanent în vedere timpul de încărcare al paginii.

14. Test

51. *Ce etichetă folosim pentru a insera o imagine în pagină?*

a) `<IMG>`
b) `<A>`
c) `<PIC>`

52. *Atributul* `src` *serveşte la:*

a) poziţionarea imaginii în pagină
b) stabilirea numelui şi locaţiei unde se află imaginea
c) crearea unei legături prin intermediul imaginii

53. *Atributele* `width` *şi* `height` *ale etichetei* `<IMG>` *sunt folosite pentru:*

a) dimensionarea textului
b) alinierea imaginii cu textul
c) dimensionarea imaginii

54. *Ce efect are absenţa atributelor* `width` *şi* `height` *din eticheta* `<IMG>`*?*

a) Nu se afişează imaginea.
b) Imaginea se afişează pe toata suprafaţa ferestrei browserului.
c) Browserul aşteaptă încărcarea completă a imaginii înainte de a începe afişarea textului.

55. *Care dintre atributele următoare permit afişarea textului în jurul imaginii?*

a) `top`
b) `bottom`
c) `middle`

56. *Care dintre construcţiile următoare este folosită pentru a stabili ca fond al paginii imaginea rice.gif?*

a) `<IMG background="rice.gif">`
b) `<IMG src="rice.gif" background>`
c) `<BODY background="rice.gif">`

57. *Ce realizează atributul* `alt` *în următoarea construcţie?*

```
<IMG src="dog.gif" alt="catel">
```

a) Afişează cuvântul `catel` lângă imagine.
b) Afişează cuvântul `catel` în spaţiul rezervat imaginii.

c) Atribuie fişierului *dog.gif* numele catel.

58. *Care dintre următoarele construcţii plasează imaginea în stânga şi aliniază textul în jurul imaginii?*

a) `<IMG src="moon.gif" wrap="left">`
b) `<IMG src="moon.gif" align="left">`
c) `<IMG src="moon.gif> align="wrap">`

59. *Care dintre următoarele construcţii are drept efect folosirea imaginii sign.gif ca legătură?*

a) `<A href="semn.html"></A><IMG src="sign.gif>`
b) `<A src="semn.html><IMG href="sign.gif"></A>`
c) `<A href="semn.html><IMG src="sign.gif"></A>`

60. *Pentru a include în pagină o imagine video se foloseşte următoarea construcţie:*

a) `<IMG src="URL_imagine">`
b) `<IMG dynsrc="URL_imagine">`
c) `<IMG video="URL_imagine">`

Imagini

1. a)
2. b)
3. c)
4. c)
5. b)
6. c)
7. b)
8. b)
9. c)
10. b)

Capitolul 8
Liste

Listele reprezintă unele dintre cele mai obişnuite elemente dintr-o pagină Web. Acestea sunt deseori folosite pentru a prezenta informaţiile în mod organizat, într-o manieră accesibilă şi uşor de parcurs.
Ele pot fi de trei tipuri:

- liste **ordonate** (marcate prin numere sau litere),
- liste **neordonate** (marcate prin cratime, buline sau alte simboluri)
- liste **de definiţii**, afişate fără nici un fel de marcaj.

În interiorul etichetelor care delimitează o listă pot fi folosite orice alte etichete HTML, cum ar fi etichete de formatare a textului, legături, imagini, etc.

1. Liste neordonate

O listă **neordonată** reprezintă o colecţie de elemente înrudite, dispuse într-o ordine oarecare. Un exemplu tipic pentru o pagină Web este o listă de link-uri spre alte documente.
Acest tip de listă este de fapt un bloc de text delimitat de etichetele **<UL> </UL>** (*unordered list*=listă neordonată). Eticheta de închidere </UL> este obligatorie.
Fiecare element al listei este introdus prin eticheta **<LI>** (*list item*). Cu toate că eticheta <LI> este o etichetă container, eticheta sa de închidere, </LI>, este opţională. Dacă nu este prezentă, la întâlnirea unei noi etichete <LI> se consideră că vechea etichetă este închisă.

Lista va fi afişată indentat faţă de restul paginii şi fiecare element al listei începe pe un rând nou. În mod prestabilit, fiecare element al listei va fi marcat (*bulleted*) cu "bulină". Exemplul 8.1 construieşte o listă neordonată, aspectul său fiind cel din
Figura 8.1.

Exemplul 8. 1

```
<HTML>
  <HEAD>
    <TITLE>liste1</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Lista neordonata</H1><HR>
    <UL>Culori
      <LI>Black
      <LI>White
      <LI>Red
      <LI>Green
      <LI>Blue
      <LI>Yellow
      <LI>Purple
      <LI>Aqua
    </UL>
  </BODY>
</HTML>
```

Etichetele <UL> şi <LI> pot avea definit atributul **type** care stabileşte caracterul afişat în faţa fiecărui element al listei. Valorile posibile al acestui atribut sunt:

- circle (cerc)
- disc (disc plin) - valoarea prestabilită
- square (patrat)

Lista din Exemplul 8.2 are atributul type setat la valoarea "square". Lista este marcată aşa cum se poate vedea în Figura 8.2

Exemplul 8. 2

```
<HTML>
  <HEAD>
    <TITLE>liste2</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Atributul type la liste neordonate</H1><HR>
    <UL type="square">Limbaje de programare
      <LI>C
      <LI>C++
      <LI>Pascal
      <LI> Basic
      <LI>Perl
```

```
    </UL>
  </BODY>
</HTML>
```

Setarea atributului `type` pentru un item al listei înlocuieşte tipul de marcaj cu tipul specificat pentru acel item.
În Figura 8.3 puteţi observa efectul setării atributului `type` pentru un item individual la listei.

Exemplul 8. 3

```
<HTML>
  <HEAD>
    <TITLE>liste3</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Atributul type la itemul unei liste </H1><HR>
    <UL type="square">Limbaje de programare
      <LI>C
      <LI>C++
      <LI>Pascal
      <LI type="circle">Basic
      <LI>Perl
    </UL>
  </BODY>
</HTML>
```

Forme particulare de liste neordonate

- **Lista de directoare -** este o listă introdusă prin eticheta **<DIR> </DIR>.** Eticheta a fost iniţial utilizată pentru alcătuirea listelor de fişiere. Multe browsere nu fac nici o diferenţă între etichetele <DIR> şi <UL>, efectul lor fiind acelaşi.

- **Lista de meniuri** - utilizează eticheta **<MENU>**. Unele browsere afişează lista doar în format uşor diferit faţă de listele neordonate, altele însă folosesc chiar un fel de meniu grafic de tip *pull-down* pentru afişarea acestor liste.

2. Liste ordonate

O listă **ordonată** este un bloc de text delimitat de etichetele **<OL> </OL>** (*ordered list* - listă ordonată), eticheta de închidere fiind obligatorie.
Fiecare element al listei este iniţiat de eticheta **<LI>** (*list item*). Ca şi în cazul listelor neordonate, lista va fi indentată faţă de restul paginii Web şi fiecare element al listei va începe pe o linie nouă.
Diferenţa faţă de listele neordonate este aceea că în acest caz marcarea elementelor se face prin cifre, nu prin simboluri.
Exemplul 8.4 afişează o listă ordonată, marcată cu cifre arabe aspectul ei fiind cel din Figura 8.4.

Exemplul 8. 4

```
<HTML>
  <HEAD>
    <TITLE>liste4</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Lista ordonata</H1><HR>
    <OL>Castigatorii concursului sunt:
      <LI>Popescu Maria
      <LI>Ionescu Ion
      <LI>Bratu Ana
```

```
    </OL>
  </BODY>
</HTML>
```

Pentru etichetele <OL> şi <LI> se poate seta atributul **type,** care stabileşte tipul de caractere utilizate pentru ordonarea listei. Valorile posibile ale acestui atribut sunt:

A - pentru ordonare de tipul A , B , C...(litere mari)
a - pentru ordonare de tipul a , b , c...(litere mici)
I - pentru ordonare de tipul I , II , III , IV...(cifre romane mari)
i - pentru ordonare de tipul i , ii , iii , iv...(cifre romane mici)
1 - pentru ordonare de tipul 1 , 2 , 3...(cifre arabe - opţiune prestabilită)

De asemenea eticheta <OL> poate avea setat atributul **start,** care stabileşte valoarea iniţială a secvenţei de ordonare. Valoarea acestui atribut trebuie să fie un număr întreg pozitiv.

Exemplul 8.5 construieşte o listă numerotată cu cifre romane mari începând de la poziţia a treia. Aspectul paginii care conţine această listă este redat în Figura 8.5.

Exemplul 8. 5

```
<HTML>
  <HEAD>
    <TITLE>liste5</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Atributul type la liste ordonate</H1><HR>
    <OL type="I" start="3">Manuale
      <LI>Istorie
      <LI>Geografie
      <LI>Biologie
      <LI>Chimie
    </OL>
  </BODY>
</HTML>
```

Ca şi în cazul listelor neordonate, setarea atributului type pentru un anumit item al listei înlocuieşte tipul de numerotare stabilit pentru întreaga listă cu tipul specificat pentru acel element.

3. Imbricarea listelor

În cadrul unei liste, fie ea ordonată sau neordonată, se pot include alte liste, procedeu numit **imbricare**. Fiecare listă nou inclusă în precedenta se va afişa indentat faţă de nivelul listei anterioare.
Listele pot fi imbricate în toate modurile dorite, ca în Exemplul 8.6. Figura 8.6 redă aspectul listei construite în acest exemplu.

Exemplul 8. 6

```
<HTML>
  <HEAD>
    <TITLE>liste6</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Liste imbricate</H1><HR>
    <UL>
      <LI>Bulina 1
```

```
        <OL>
          <LI>Numarul 1
          <LI>Numarul 2
        </OL>
      <LI>Bulina 2
      <LI>Bulina 3
      <UL type="square">
        <LI>Patrat 1
        <LI>Patrat 2
        <LI>Patrat 3
      </UL>
      <LI>Bulina 4
    </UL>
  </BODY>
</HTML>
```

4. Liste de definiţii

Listele de **definiţii** reprezintă un tip special de liste în care elementele listei nu sunt nici numerotate (ca în listele ordonate), nici marcate prin buline (ca în listele neordonate) şi care prezintă două nivele de indentare.

Listele de definiţii sunt blocuri de text incluse între etichetele **<DL> </DL>** (*definition list*), eticheta de închidere fiind obligatorie. Fiecare element al listei este introdus prin eticheta **<DT>** (*definition term*). Fiecare element introdus prin eticheta <DT> conţine la rândul său un număr de elemente care îl definesc, introduse prin eticheta **<DD>** (*definition description*).

Ca şi în cazul etichetei <LI>, etichetele de închidere </DT> şi </DD> sunt opţionale. În Exemplul 8.7 este construită o listă de definiţii.

Exemplul 8. 7

```
<HTML>
  <HEAD>
    <TITLE>liste7</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Liste de definitii</H1><HR>
    <DL>
      <DT>Iarna
        <DD>Ninge
        <DD>E frig
        <DD>Ziua e mai scurta decât noaptea
      <DT>Primavara
        <DD>Natura se trezeste la viata
        <DD>Infloresc pomii
        <DD>Se intorc pasarile calatoare
      <DT>Vara
        <DD>Totul e verde
        <DD>E foarte cald
        <DD>Ziua e mai lunga decât noaptea
      <DT>Toamna
        <DD>Se coc fructele
        <DD>Se strange recolta
        <DD>Cad frunzele
    </DL>
  </BODY>
</HTML>
```

Aspectul listei de definiţii de mai sus este cel din Figura 8.7.

Etichetele <DL> şi </DL> marchează începutul şi sfârşitul listei, termenii care fac parte din listă (Iarna, Primăvara, Vara, Toamna) sunt introduşi prin eticheta <DT> iar definiţiile termenilor, prin etichetele <DD>.

Se pot realiza liste ale căror elemente să fie link-uri, imagini sau blocuri de text. În Exemplul 8.8, elementele listei sunt imagini, aşa cum se poate vedea în Figura 8.8.

Exemplul 8. 8

```
<HTML>
  <HEAD>
    <TITLE>liste8</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Liste de imagini</H1><HR>
    <DL>
      <DT>Flori
        <DD>
        <IMG src="../Imagini/rose1.jpg" width="120" height="120">
        <DD>
        <IMG src="../Imagini/orangerose.gif" width="120" height="120">
      <DT>Texturi
        <DD>
        <IMG src="../Imagini/silk.jpg" width="120" height="120">
        <DD>
        <IMG src="../Imagini/lace.gif" width="120" height="120">
    </DL>
  </BODY>
</HTML>
```

Exemplul 8.9 creează două liste imbricate de legături. Aspectul paginii care conţine această listă este redat în Figura 8.9. Puteţi testa direct funcţionarea listei dacă aţi salvat fişierele HTML construite la capitolele anterioare sub numele specificat în blocul <TITLE> al fiecărui document. În cazul când aţi preferat alte denumiri pentru fişiere, puteţi schimba numele fişierelor prezente în cadrul exemplului cu propriile dumneavoastră fişiere.

Exemplul 8. 9

```
<HTML>
  <HEAD>
    <TITLE>liste9</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Liste de legaturi</H1><HR>
    <OL type="I">
      <LI>Culori
      <OL>
        <LI><A href="culori1.html">Culoarea fundalului</A>
        <LI><A href="culori2.html">Culoarea textului</A>
        <LI><A href="culori3.html">Culoarea legaturilor</A>
      </OL>
      <LI>Fonturi
      <OL>
        <LI><A href="fonturi1.html">Eticheta FONT</A>
        <LI><A href="fonturi2.html">Eticheta BASEFONT</A>
        <LI><A href="fonturi4.html">Etichete de accentuare a textului</A>
      </OL>
    </OL>
  </BODY>
```

```
</HTML>
```

5. Rezumat

Într-o pagină Web pot fi introduse diverse tipuri de liste:

- Liste neordonate prin eticheta `<UL> </UL>`, marcate prin simboluri de tip buline, cercuri sau pătrate
- Liste ordonate prin eticheta `<OL> </OL>` marcate prin cifre arabe, cifre romane mari sau mici, litere mari sau mici
- Liste de definiţii prin eticheta `<DL> </DL>` nemarcate

Elementele listelor ordonate şi neordonate se introduc prin eticheta `<LI>`.

Elementele listelor de definiţii se introduc prin etichetele `<DT>` şi `<DD>`.

Listele pot conţine orice tip de elemente: text, imagini, link-uri sau alte liste (în acest caz obţinându-se liste imbricate).

6. Test

61. *Eticheta* `<UL>` *se foloseşte pentru a iniţia:*

a) o listă ordonată
b) o listă neordonată
c) două liste imbricate

62. *Care dintre următoarele afirmaţii este falsă?*

a) Eticheta `<LI>` serveşte pentru a introduce fiecare element al unei liste ordonate.
b) Eticheta `<LI>` serveşte pentru a introduce fiecare element al unei liste neordonate.
c) Eticheta `<LI>` serveşte pentru a introduce fiecare element al unei liste de definiţii.

63. *În construcţia* `<UL type="square">` *ce rol are atributul* `type`*?*

a) Construieşte o listă neordonată.
b) Stabileşte caracterul care va fi afişat în faţa fiecărui element al listei.
c) Construieşte o listă care are ca elemente imagini.

64. *O listă ordonată este introdusă prin eticheta:*

a) `<UL>`
b) `<OL>`
c) `<DL>`

65. *Ce rol are atributul* `type` *în următoarea construcţie:* `<OL type="a">`*?*

a) Construieşte o listă de litere care începe cu caracterul `"a"`.
b) Construieşte o listă ordonată.
c) Stabileşte faptul că elementele listei vor fi marcate cu litere mici.

66. *Ce este incorect în exemplul următor?*

```
<UL>
```

```
    <LI>Limbaje de programare
    <OL>
      <LI>C++
      <LI>Perl
      <LI>Java
    </OL>
    <LI>Sisteme de operare>
    <OL>
      <LI>MSDos
      <LI>Unix
    </UL>
  </OL>
```

a) Nu este permisă imbricarea listelor neordonate cu liste ordonate.
b) Este absentă eticheta de închidere `</LI>`.
c) Etichetele `<UL>` şi `<OL>` nu sunt închise corect.

67. *O listă de definiţii se introduce prin eticheta:*

a) `<UL>`
b) `<DT>`
c) `<DL>`

68. *Care este ordinea corectă în care se introduc elementele unei liste de definiţii?*

a) a) `<DL> <DT> <DD>`
b) b) `<DD> <DT> <DL>`
c) c) `<DT> <DL> <DD>`

69. *Ce realizează următoarea construcţie?*

```
<UL>
  <LI><IMG src="pic1.gif">
  <LI><IMG src="pic2.gif">
</UL>
```

a) Construieşte o listă de imagini marcate prin buline.
b) Construieşte o listă de link-uri către imaginile specificate.
c) Nimic, deoarece este incorect ca elementele unei liste să fie imagini.

70. *Cum sunt marcate elementele unei liste de definiţii?*

a) prin buline
b) prin numere
c) nu sunt marcate

Liste

1. b)
2. c)
3. b)
4. b)
5. c)
6. c)
7. c)
8. a)
9. a)

10. c)

Capitolul 9
Tabele

Tabelul este un element structural de bază în alcătuirea unei pagini Web. În cadrul unui tabel pot fi incluse oricare dintre elementele care fac parte din corpul unui document HTML: text, imagini, linii orizontale, titluri, legături, alte tabele, etc. Tabelele constituie un instrument extrem de util de organizare a paginii Web, realizând alinierea elementelor şi plasarea lor în locurile dorite de designerul paginii.

Un tabel este o grilă dreptunghiulară formată din **linii** şi **coloane**. Caseta formată la intersecţia unei linii cu o coloană se numeşte **celulă**.
O **linie** a tabelului este formată dintr-un şir de celule aliniate pe orizontală, iar o **coloană** este formată dintr-un şir de celule aliniate pe verticală.

Celulele tabelului conţin date (text, imagini, link-uri), fiecare celulă având propriile opţiuni pentru culoarea fondului, culoarea textului, alinierea textului etc.

1. Crearea unui tabel

Pentru a insera un tabel într-un document HTML se folosesc etichetele corespondente **<TABLE> </TABLE>**. Eticheta <TABLE> este o etichetă container, deci eticheta de final este obligatorie. Absenţa ei face ca tabelul să nu fie afişat corect.

Pentru a insera o linie într-un tabel se folosesc etichetele **<TR> </TR>** (*table row*). Folosirea etichetei de închidere </TR> este opţională.

Aşa cum spuneam, fiecare linie de tabel este formată din mai multe celule ce conţin date. O celulă de date se introduce cu eticheta **<TD> </TD>** (*table data*). Eticheta de închidere </TD> este de asemenea opţională.

În Exemplul 9.1 am construit un tabel format din patru linii, fiecare linie având câte două celule de date. Aspectul tabelului este redat în Figura 9.1.

Exemplul 9. 1

```
<HTML>
  <HEAD>
    <TITLE>tabele1</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Crearea unui tabel</H1><HR>
    <TABLE>
      <TR>
        <TD>celula 11
        <TD>celula 11
      <TR>
        <TD>celula 21
        <TD>celula 22
      <TR>
        <TD>celula 31
        <TD>celula 32
      <TR>
```

```
        <TD>celula 41
        <TD>celula 42
    </TABLE>
  </BODY>
</HTML>
```

În mod prestabilit, un tabel nu are chenar vizibil. Pentru a adăuga un chenar unui tabel, se utilizează atributul **`border`** al etichetei `<TABLE>`. Acest atribut poate primi ca valoare orice număr întreg (inclusiv 0) şi reprezintă grosimea în pixeli a chenarului tabelului.
Daca atributul `border` nu este urmat de o valoare atunci chenarul tabelului va avea o grosime prestabilită egală cu 1 pixel, iar o valoare egală cu 0 a atributului `border` semnifică absenţa chenarului.
Când atributul `border` are o valoare nenulă chenarul tabelului are un aspect tridimensional.

În Exemplul 9.2 este construit un tabel cu chenar, aşa cum se observă în Figura 9.2. Pentru a testa funcţionarea atributului `border`, înlocuiţi în exemplul de mai jos valoarea "4" şi cu alte valori. Nu uitaţi ca după fiecare modificare să salvaţi fişierul şi să apăsaţi butonul *Refresh/Reload* al browserului.

Exemplul 9. 2

```
<HTML>
  <HEAD>
    <TITLE>tabele2</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Chenarul unui tabel</H1><HR>
    <TABLE border="4">
      <TR>
        <TD>celula 11
        <TD>celula 12
      <TR>
        <TD>celula 21
        <TD>celula 22
      <TR>
        <TD>celula 31
        <TD>celula 32
      <TR>
        <TD>celula 41
        <TD>celula 42
    </TABLE>
  </BODY>
</HTML>
```

Celulele unui tabel pot conţine şi alte elemente în afară de text: imagini, legături, formulare, etc.
Exemplul 9.3 construieşte un tabel cu două linii şi două coloane, celulele tabelului având drept conţinut imagini. Aspectul tabelului este cel din Figura 9.3.

Exemplul 9. 3

```
<HTML>
  <HEAD>
    <TITLE>tabele3</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Tabel care contine imagini</H1><HR>
    <TABLE border="4">
```

```
      <TR>
        <TD><IMG src="../Imagini/donut.gif">
        <TD><IMG src="../Imagini/gift.gif">
      <TR>
        <TD><IMG src="../Imagini/invest.gif">
        <TD><IMG src="../Imagini/globe.gif">
    </TABLE>
  </BODY>
</HTML>
```

2. Alinierea tabelului în pagină

Pentru a alinia un tabel într-o pagina Web se utilizează atributul **`align`** al etichetei `<TABLE>`, cu următoarele valori posibile:

- `left` (valoarea prestabilită) - textul care urmează după punctul de inserare al tabelului va fi dispus în partea dreaptă a tabelului.
- `center` - textul care urmează după punctul de inserare al tabelului va fi afişat pe toată lăţimea paginii, imediat sub tabel.
- `right` - textul care urmează după punctul de inserare al tabelului va fi dispus în partea stângă a tabelului.

În Exemplul 9.4 este ilustrată construirea unui tabel aliniat la dreapta. Pagina va avea aspectul din Figura 9.4.

Exemplul 9. 4

```
<HTML>
  <HEAD>
    <TITLE>tabele4</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Alinierea tabelului in pagina</H1><HR>
    <TABLE border="3" align="right">
      <TR>
        <TD>celula 11
        <TD>celula 12
      <TR>
        <TD>celula 21
        <TD>celula 22
    </TABLE>
    Acest text este plasat in stanga tabelului
  </BODY>
</HTML>
```

Distanţa dintre tabel şi celelalte elemente din pagina Web poate fi stabilită cu ajutorul atributelor **`hspace`** şi **`vspace`** al etichetei `<TABLE>`.
Valoarea atributului `hspace` poate fi orice număr pozitiv, inclusiv 0, şi reprezintă distanţa pe orizontală dintre tabel şi celelalte elemente ale paginii Web.
Analog, valoarea atributului `vspace` reprezintă distanţa pe verticală dintre tabel şi celelalte elemente ale paginii. Atributele `hspace` şi `vspace` sunt recunoscute numai de browserele Netscape.
Exemplul 9.5 ilustrează utilitatea celor două atribute.

Exemplul 9. 5

```
<HTML>
  <HEAD>
    <TITLE>tabele5</TITLE>
  </HEAD>
  <BODY>
```

```
    <H1 align="center">Spatierea tabelului fata de restul textului
    </H1><HR>
    Tabelul de mai jos este spatiat pe verticala fata de acest text cu 15
    pixeli. Tabelul de mai jos este spatiat pe verticala fata de acest
    text cu 15 pixeli. Tabelul de mai jos este spatiat pe verticala fata
    de acest text cu 15 pixeli<BR>
    <TABLE border align="left" vspace="15" hspace="10">
      <TR>
        <TD>celula 11
        <TD>celula 12
      <TR>
        <TD>celula 21
        <TD>celula 22
    </TABLE>
    Acest text este plasat in dreapta tabelului la distanta de 10 pixeli
    pe orizontala fara de tabel. Acest text este plasat in dreapta
    tabelului la distanta de 10 pixeli pe orizontala fara de tabel. Acest
    text este plasat in dreapta tabelului la distanta de 10 pixeli pe
    orizontala fara de tabel.
  </BODY>
</HTML>
```

Pentru a observa efectul celor două atribute editaţi acest exemplu şi vizualizaţi-l cu un browser Netscape.
Atributele nefiind recunoscute de Internet Explorer, aspectul paginii la vizualizarea cu acest browser va fi cel din Figura 9.5.

3. Dimensionarea unui tabel

Dimensiunile unui tabel - lăţimea şi înălţimea - pot fi stabilite exact prin intermediul atributelor **`width`** şi **`height`** ale etichetei `<TABLE>`.

Valorile acestor atribute pot fi:

- numere întregi pozitive reprezentând lăţimea respectiv înălţimea în pixeli a tabelului
- numere întregi între 1 şi 100, urmate de semnul %, reprezentând procente din lăţimea şi înălţimea totală a paginii.

Iată un exemplu de tabel cu înălţimea de 200 de pixeli şi lăţimea egală cu 50% din lăţimea paginii (Exemplul 9.6)

Exemplul 9. 6

```
<HTML>
  <HEAD>
    <TITLE>tabele6</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Dimensionarea unui tabel</H1><HR>
    Tabel cu lungimea de 50% din pagina si inaltimea de 200 de pixeli<P>
    <TABLE border width="50%" height="200">
      <TR>
        <TD>celula 11
        <TD>celula 12
      <TR>
        <TD>celula 21
        <TD>celula 22
    </TABLE>
  </BODY>
</HTML>
```

Aspectul tabelului construit în documentul de mai sus este cel din Figura 9.6.

Dimensiunile unui tabel includ şi valoarea dată atributului `border`. De exemplu, un tabel cu o singură linie şi o singură coloană, cu dimensiunile `width="100"`, `height="50"` şi având atributul `border` setat la valoarea 10, va avea drept spaţiu util 80 de pixeli pe lăţime şi 30 de pixeli pe înălţime.

Una dintre cele mai frecvente utilizări a tabelelor este poziţionarea unui text într-o anumită zonă a paginii, prin realizarea unui tabel fără chenar (`border="0"`) cu o singură linie şi o singură coloană, ca în Exemplul 9.7.

Exemplul 9. 7

```
<HTML>
  <HEAD>
    <TITLE>tabele7</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Pozitionarea unui text</H1>
    <TABLE border="0" width="60%" height="100%" align="center">
      <TR>
        <TD>
          Acest text este pozitionat in centrul paginii. Acest text este
          pozitionat in centrul paginii. Acest text este pozitionat in
          centrul paginii. Acest text este pozitionat in centrul paginii.
          Acest text este pozitionat in centrul paginii.
    </TABLE>
  </BODY>
</HTML>
```

În Figura 9.7 este redat aspectul tabelului construit mai sus.

4. Spaţierea celulelor unui tabel

Distanţa dintre două celule vecine se defineşte cu ajutorul atributului **`cellspacing`** al etichetei `<TABLE>`.
Valorile acestui atribut pot fi numere întregi pozitive, inclusiv 0, şi reprezintă distanţa în pixeli dintre două celule vecine. Valoarea prestabilită a atributului `cellspacing` este de 2 pixeli.
Exemplul 9.8 ilustrează funcţionarea atributului `cellspacing`. Aspectul tabelului este cel din Figura 9.8. Puteţi modifica valoarea atributului `cellspacing` pentru a observa cum se spaţiază celulele în funcţie de valorile pe care le daţi.

Exemplul 9. 8

```
<HTML>
  <HEAD>
    <TITLE>tabele8</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Spatierea celulelor</H1><HR>
    <TABLE border="3" cellspacing="10">
      <TR>
        <TD>Maria
        <TD>Bogdan
      <TR>
        <TD>Alexandru
        <TD>Irina
    </TABLE>
  </BODY>
</HTML>
```

Distanţa dintre marginea unei celule şi conţinutul ei poate fi definită cu ajutorul atributului **cellpadding** al etichetei <TABLE>. Valorile acestui atribut pot fi numere întregi pozitive, şi reprezintă distanţa în pixeli dintre marginile celulei şi conţinutul ei. Valoarea prestabilită a atributului cellpadding este 1 pixel.

Exemplul 9.9 construieşte un tabel în care distanţa dintre marginea celulelor şi conţinutul lor este de 20 de pixeli. Aspectul acestui tabel este cel din Figura 9.9.

Exemplul 9. 9

```
<HTML>
  <HEAD>
    <TITLE>tabele9</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center"> Spatierea textului in celule</H1><HR>
    <TABLE border cellpadding="20">
      <TR>
        <TD>Maria
        <TD>Bogdan
      <TR>
        <TD>Alexandru
        <TD>Irina
    </TABLE>
  </BODY>
</HTML>
```

5. Dimensionarea celulelor unui tabel

Dimensiunile unei celule de tip <TD> sau de tip <TH> (vezi mai jos eticheta <TH>) pot fi stabilite exact cu ajutorul a două atribute ale acestor etichete: **width** pentru lăţime şi **height** pentru înălţime. Valorile posibile ale acestor atribute sunt:

- numere întregi pozitive (inclusiv 0) reprezentând dimensiunea în pixeli a lăţimii, respectiv a înălţimii unei celule
- procente din lăţimea , respectiv înălţimea tabelului.

În Exemplul 9.10 am dimensionat celula 11 la lăţimea de 20% din lăţimea tabelului şi înălţimea egală cu 75% din înălţimea lui. Aşa cum se poate observa din Figura 9.10 dimensionarea individuală a unei celule va afecta dimensionarea tuturor celulelor din linia şi coloana din care face parte celula respectivă. Se observă că celula 12 are lăţimea egală cu restul de 80% din lăţimea tabelului. Celula 21 aflată pe aceeaşi coloana cu celula dimensionată are aceleaşi dimensiuni ca şi ea.

Exemplul 9. 10

```
<HTML>
  <HEAD>
    <TITLE>tabele10</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Dimensionarea celulelor</H1><HR>
    Tabel cu celule dimensionate individual<P>
    <TABLE border>
      <TR>
        <TD width="20%" height="75%">celula 11
        <TD>celula 12
      <TR>
        <TD>celula 21
```

```
        <TD>celula 22
    </TABLE>
  </BODY>
</HTML>
```

6. Alinierea conţinutului unei celule

Alinierea pe orizontală a conţinutului unei celule se face cu ajutorul atributului **`align`** care poate lua valorile:

- `left` - la stânga
- `center` - centrat , valoarea prestabilită
- `right` - la dreapta

Alinierea pe verticală a conţinutului unei celule se face cu ajutorul atributului **`valign`** care poate lua valorile:

- `baseline` - la bază
- `bottom` - jos
- `middle` - la mijloc, valoarea prestabilită
- `top` - sus

Aceste atribute pot fi ataşate atât etichetei `<TR>` pentru a defini alinierea tuturor celulelor unei linii, cât şi etichetelor `<TD>` şi `<TH>` (vezi mai jos eticheta `<TH>`) pentru a stabili alinierea textului într-o singură celulă. În Exemplul 9.11 este ilustrată funcţionarea atributelor `align` şi `valign`.
Textul din celulele primei, celei de-a doua şi celei de-a patra linii a fost aliniat prin atributul `align` asociat liniei, iar textul din celulele liniei a treia a fost aliniat prin atributul `align` asociat fiecărei celule în parte. Aspectul tabelului este cel din Figura 9.11.

Exemplul 9. 11

```
<HTML>
  <HEAD>
    <TITLE>tabele11</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center"> Alinierea continutului celulelor</H1><HR>
    <TABLE border width="50%" height="80%">
      <TR align="right">
        <TD>dreapta
        <TD>dreapta
      <TR align="center">
        <TD>centru
        <TD>centru
      <TR>
        <TD valign="top">sus
        <TD valign="bottom">jos
      <TR align="left">
        <TD>stanga
        <TD>stanga
    </TABLE>
  </BODY>
</HTML>
```

Dacă este prezent un atribut de aliniere ataşat etichetei `<TR>` şi pe linia respectivă o anumită celulă are propriul său atribut de aliniere, atributul de aliniere asociat etichetei `<TD>` are prioritate faţă de cel asociat etichetei `<TR>`.

7. Definirea culorilor pentru un tabel

Culoarea de fond a unui tabel se stabileşte cu ajutorul atributului **bgcolor**, care poate fi ataşat după cum urmează:

- întregului tabel prin eticheta <TABLE>,
- unei linii prin eticheta <TR>
- unei celule de date prin eticheta <TD>

Valorile pe care le poate primi atributul bgcolor sunt cele cunoscute pentru culori.
Dacă într-un tabel sunt definite mai multe atribute bgcolor, atunci prioritatea este, în ordine descrescătoare, următoarea:

1. <TD>
2. <TR>
3. <TABLE> (prioritatea cea mai mica)

În Exemplul 9.12 este ilustrată folosirea atributului bgcolor.

Exemplul 9. 12

```
<HTML>
  <HEAD>
    <TITLE>tabele12</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Culori in tabel</H1><HR>
    <TABLE border="5" bgcolor="lime">
      <TR>
        <TD>celula 11 verde
        <TD bgcolor="red">celula 12 rosu
      <TR bgcolor="blue">
        <TD>celula 21 albastru
        <TD bgcolor="yellow">celula 22 galben
      <TR bgcolor="cyan">
        <TD>celula 31 cyan
        <TD>celula 32 cyan
      <TR>
        <TD>celula 41 verde
        <TD bgcolor="white">celula 42 alb
    </TABLE>
  </BODY>
</HTML>
```

Din Figura 9.12 puteţi observa cum funcţionează prioritatea atributului bgcolor. Astfel, culoarea întregului tabel a fost stabilită prin eticheta:

```
<TABLE bgcolor="lime">
```

Dacă o linie nu are stabilită nici o alta culoare, culoarea liniei este verde deschis. Pentru linia a doua a tabelului este stabilită culoarea albastru:

```
<TR bgcolor="blue">.
```

Celulele care fac parte din aceasta linie şi care nu au stabilită o alta culoare, vor fi albastre (celula 21). Celelalte celule vor avea culoarea stabilită explicit pentru ele (celula 22 va fi galbenă):

```
<TD bgcolor="yellow">
```

Un tabel poate avea drept fundal nu numai o culoare, ci şi o imagine. Acest efect se obţine folosind atributul **background** al etichetei <TABLE>, atributul primind ca valoare adresa URL a imaginii. Atributul background poate fi ataşat şi unei linii, stabilind fundalul tuturor celulelor din linia respectivă precum şi unei celule individuale, în acest caz stabilind fundalul acelei celule.

Exemplul 9.13 ilustrează acest atribut, tabelul având aspectul redat în Figura 9.13.

Exemplul 9. 13

```
<HTML>
  <HEAD>
    <TITLE>tabele13</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Fundalul unui tabel</H1><HR>
    <TABLE border="5" width="200" height="120"
    background="../Imagini/lace.gif>
      <TR>
        <TD>celula 11
        <TD>celula 12
      <TR>
        <TD>celula 21
        <TD>celula 22
    </TABLE>
  </BODY>
</HTML>
```

Culorile chenarului unui tabel se pot stabili folosind atributele:

bordercolor - permite stabilirea culorii pentru chenarul unui tabel
bordercolorlight - permite stabilirea culorii marginilor din stânga şi de sus ale tabelului
bordercolordark - permite stabilirea culorii marginilor din dreapta şi de jos ale tabelului

Culoarea textului din fiecare celulă se poate stabili cu ajutorul expresiei:

```
<FONT color="#RGB sau nume_culoare">text</FONT>.
```

Exemplul 9.14 ilustrează folosirea acestor atribute, aspectul tabelului fiind cel din Figura 9.14.

Exemplul 9. 14

```
<HTML>
  <HEAD>
    <TITLE>tabele14</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Culorile chenarului unui tabel</H1><HR>
    <TABLE border="10" bordercolordark="blue" bordercolorlight="cyan">
      <TR>
        <TD><FONT color="green">celula 11</FONT>
        <TD><FONT color="blue">celula 12</FONT>
        <TD><FONT color="red">celula 13</FONT>
      <TR>
        <TD><FONT color="teal">celula 21</FONT>
        <TD><FONT color="magenta">celula 22</FONT>
        <TD><FONT color="lime">celula 23</FONT>
```

```
    </TABLE>
  </BODY>
</HTML>
```

Am setat grosimea chenarului tabelului la valoarea de 10 pixeli, pentru a avea un aspect tridimensional mai pronunţat.
Cu ajutorul atributului `bordercolordark` am stabilit culoarea părţii "umbrite" a chenarului (marginile de jos şi din dreapta) iar cu ajutorul atributului `bordercolorlight`, culoarea părţii "luminate" a acestuia (marginile de sus şi din stânga) iar textul din celule l-am scris cu culori diferite, folosind eticheta `<FONT>`.
Culorile stabilite pentru chenarul exterior al tabelului sunt folosite şi pentru a colora liniile despărţitoare dintre celulele tabelului. Pentru acestea partea "umbrită" este formată din muchiile de sus şi din stânga iar partea "luminată" este formată din muchiile de jos şi din dreapta.
Dacă dorim să colorăm separat muchiile care despart liniile sau celulele tabelului putem asocia atributele `bordercolor`, `bordercolordark` şi `bordercolorlight` etichetelor `<TR>` şi `<TD>`.

Atenţie!

Atributele `bordercolor`, `bordercolordark` şi `bordercolorlight` nu sunt recunoscute de browserul Nescape.

Vizualizaţi pagina de mai sus cu un browser Netscape pentru a observa aspectul ei.

8. Titlul unui tabel

Unui tabel i se poate ataşa un titlu cu ajutorul etichetei **`<CAPTION>`** (*table caption*=titlu tabel).
Această etichetă trebuie plasată în interiorul etichetelor `<TABLE> </TABLE>`, dar **nu** în interiorul etichetelor `<TR>` sau `<TD>`.
Titlul unui tabel poate fi aliniat cu ajutorul atributului **`align`** al etichetei `<CAPTION>` care poate lua una dintre valorile:

- `bottom` - sub tabel
- `top` - deasupra tabelului
- `left` - deasupra, la stânga tabelului
- `right` - deasupra, la dreapta tabelului

În Exemplul 9.15 titlul tabelului este aliniat deasupra sa. Aspectul paginii este cel din Figura 9.15.

Exemplul 9. 15

```
<HTML>
  <HEAD>
    <TITLE>tabele15</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Titlul unui tabel</H1><HR>
    <TABLE border><CAPTION align="top">Masini
      <TR>
        <TD>Mercedes
        <TD>Citroen
        <TD>Jaguar
      <TR>
        <TD>BMW
        <TD>Volvo
        <TD>Renault
```

```
    </TABLE>
  </BODY>
</HTML>
```

9. Capul de tabel

Un tabel poate avea celule cu semnificaţia de cap de tabel. Aceste celule sunt introduse de eticheta **<TH>** (*tabel header*=cap de tabel) în loc de <TD>.
Toate atributele care pot fi ataşate etichetei <TD> pot fi de asemenea ataşate şi etichetei <TH>. Conţinutul celulelor definite cu <TH> este scris cu caractere aldine şi centrat.

Exemplul 9.16 ilustrează modul în care se pot insera celule cu rol de cap de tabel. Puteţi observa din exemplu că elementele cu rol de cap de tabel pot fi plasate atât pe orizontală cât şi pe verticală. Aspectul tabelului este redat în Figura 9.16.

Exemplul 9. 16

```
<HTML>
  <HEAD>
    <TITLE>tabele16</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Capul de tabel</H1><HR><P>
    <TABLE border><CAPTION align="bottom">Preturi masini
      <TR>
        <TH>Pret
        <TH>Citroen
        <TH>Jaguar
        <TH>BMW
        <TH>Volvo
      <TR>
        <TH>In dolari
        <TD>5000
        <TD>100000
        <TD>50000
        <TD>80000
      <TR>
        <TH>In lei
        <TD>5
        <TD>100
        <TD>50
        <TD>80
    </TABLE>
  </BODY>
</HTML>
```

10. Tabele de forme oarecare

Există situaţii când dorim ca o celulă să se extindă peste celulele vecine, pe orizontală sau pe verticală. În acest mod se obţine o singură celulă cu suprafaţa egală cu suma suprafeţelor celulelor iniţiale.

Acest lucru se poate realiza cu ajutorul atributelor **colspan** şi **rowspan** ale etichetelor <TD> şi <TH>.
Astfel:

colspan - realizează extinderea unei celule peste celulele din dreapta ei. Valoarea atributului determină numărul de celule care se unifică.

rowspan - realizează extinderea unei celule peste celulele de sub ea. Valoarea atributului determină numărul de celule care se unifică.

Sunt posibile extinderi simultane ale unei celule pe orizontală şi pe verticală. În acest caz, în etichetele <TD> sau <TH> vor fi prezente ambele atribute colspan şi rowspan. Exemplul 9.17 ilustrează modul cum se realizează un tabel cu celule unificate.

Exemplul 9. 17

```
<HTML>
  <HEAD>
    <TITLE>tabele17</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Tabele cu forma oarecare</H1><HR>
    <TABLE border>
      <TR>
        <TD rowspan="3">celula 11<BR>celula 21<BR>celula 31
        <TD>celula 12
        <TD colspan="2" rowspan="3">celula 13 , celula 14<BR>celula 23,
        celula 24<BR>celula 33, celula 34
      <TR>
        <TD>celula 22
      <TR>
        <TD>celula 32
      <TR>
        <TD>celula 41
        <TD colspan="3">celula 42, celula 43, celula 44
    </TABLE>
  </BODY>
</HTML>
```

Tabelul construit în acest exemplu are 4 linii şi 4 coloane. Aşa cum se vede în Figura 9.17, prin folosirea atributelor colspan şi rowspan configuraţia tabelului s-a modificat astfel: celula 11 s-a extins în jos peste celulele 21 şi 31 (<TD rowspan="3">), celula 12 a rămas nemodificată, celula 13 s-a extins atât spre dreapta peste celula 14 cât şi în jos, peste celulele 23 şi 24 (<TD colspan="2" rowspan="3">).
Celulele 22, 32 şi 41 au rămas nemodificate iar celula 42 s-a extins spre dreapta peste celulele 43 şi 44 (<TD colspan="3">.

Etichetelor <TD> şi <TH> li se poate ataşa atributul **nowrap**. El interzice ajustarea automată a lungimii unei linii de text, astfel încât în tabel pot apărea coloane cu o lăţime oricât de mare. Un tabel în care este folosit atributul nowrap este cel construit în Exemplul 9.18. Aspectul tabelului este redat în Figura 9.18.

Exemplul 9. 18

```
<HTML>
  <HEAD>
    <TITLE>tabele18</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center"> Atributul nowrap</H1><HR>
    <TABLE border>
      <TR>
        <TD>celula 11
        <TD>celula 12
      <TR>
        <TD>celula 21
        <TD nowrap>celula 22 Textul din aceasta celula este foarte lung
```

```
    </TABLE>
  </BODY>
</HTML>
```

11. Celule fără conţinut

Dacă un tabel are celule fără conţinut (celule vide), atunci aceste celule vor apărea în tabel fără un chenar de delimitare. În scopul de a afişa un chenar pentru celulele vide se poate proceda astfel:

- după eticheta <TD> se adaugă ** **
- după eticheta <TD> se adaugă **
**

Caracterul   (*no break space*) este, aşa cum am văzut la capitolul **Formatarea textului**, caracterul spaţiu. Un spaţiu introdus prin intermediul acestui caracter nu va fi ignorat de browser.

În Exemplul 9.19 este ilustrat modul în care pot fi create celule fără conţinut care să aibă totuşi chenar de delimitare. După cum observaţi din Figura 9.19, celula 23 nu are chenar de delimitare. Celulele de pe a doua linie a tabelului sunt vide dar au chenar deoarece am folosit   şi <BR>.

Exemplul 9. 19

```
<HTML>
  <HEAD>
    <TITLE>tabele19</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Celule fara continut</H1><HR>
    <TABLE border>
      <TR>
        <TD>celula 11
        <TD>celula 12
        <TD>celula 13
      <TR>
        <TD> 
        <TD><BR>
        <TD>
    </TABLE>
  </BODY>
</HTML>
```

12. Grupuri de coloane

Etichetele **<COLGROUP> </COLGROUP>** permit definirea unui grup de coloane în cadrul unui tabel. Atributele acceptate de <COLGROUP> sunt:

span - determină numărul de coloane dintr-un grup
width - determină o lăţime unică pentru coloanele din grup
align - determină un tip unic de aliniere pentru coloanele din grup

De exemplu:

```
<COLGROUP span="3" width="100"></COLGROUP>
```

Am definit astfel un grup de trei coloane, fiecare având lăţimea de 100 de pixeli.

Într-un bloc <COLGROUP>, coloanele pot avea configurări diferite dacă se utilizează eticheta **<COL>**, care admite atributele:

span - identifică acea coloană din grup pentru care se face configurarea. Dacă lipseşte, atunci coloanele sunt configurate în ordine.
width - determină lăţimea coloanei identificate prin `span`.
align - determină alinierea conţinutului coloanei identificate prin `span`.

Cel mai bine puteţi înţelege funcţionarea acestei etichete din Exemplul 9.20. Tabelul din acest exemplu are dimensiunile de 420 de pixeli lăţime şi 200 de pixeli înălţime şi este format din două linii şi patru coloane.
În tabel a fost delimitat un grup de trei coloane (primele trei coloane) prin eticheta `<COLGROUP>` iar în cadrul acestui grup au fost definite lăţimea şi alinierea textului din fiecare coloane în parte. Cea de-a patra coloană a tabelului nu face parte din grupul de coloane. După închiderea etichetei `<COLGROUP>`, celulele de date au fost introduse în mod normal, cu ajutorul etichetei `<TD>`.

Exemplul 9. 20

```
<HTML>
  <HEAD>
    <TITLE>tabele20</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Grupuri de coloane</H1><HR>
    <TABLE border width="420" height="200">
      <COLGROUP>
        <COL width="60" align="left">
        <COL width="120" align="center">
        <COL width="180" align="right">
      </COLGROUP>
      <TR>
        <TD valign="top">celula 11
        <TD>celula 12
        <TD valign="bottom">celula 13
        <TD>celula 14
      <TR>
        <TD>celula 21
        <TD>celula 22
        <TD>celula 33
        <TD>celula 34
    </TABLE>
  </BODY>
</HTML>
```

Din Figura 9.20 se poate observa că în cadrul celulei 11 este respectată alinierea la stânga a textului, deoarece ea a fost setata ca atare prin construcţia `<COL width="60" align="left">`.

În plus, în cadrul etichetei `<TD>` a fost stabilita şi alinierea pe verticala a textului din celulă, atributul valign primind valoarea "top".
În celula 12 nu este setat atributul valign astfel încât textul respectă alinierea stabilită prin construcţia `<COL width="120" align="center">`. În celula 13 alinierea pe orizontală a textului este stabilită prin construcţia `<COL width="180" align="right">` iar alinierea pe verticală este setată în cadrul etichetei `<TD>` la valoarea "bottom".

Din figură se mai observă că toate celulele care fac parte din coloanele incluse în blocul `<COLGROUP>` respectă setările făcute în cadrul acestui bloc (celulele 21, 22 şi 23). Celulele 14 şi 24 care formează coloana a patra a tabelului neinclusă în grupul de coloane,

respectă atributele de aliniere specificate în cadrul etichetei <TD>, dacă există.

Eticheta <COLGROUP> nu este recunoscută de browserele Netscape şi prin urmare nici atributele legate de dimensionarea coloanelor şi alinierea textului în interiorul lor.

13. Atribute pentru aspectul chenarului unui tabel

Atributul `frame`

Atributul **`frame`** al etichetei <TABLE> permite specificarea laturilor din chenarul unui tabel care vor fi vizibile.
Valorile posibile ale acestui atribut sunt:

- `void` - elimină toate muchiile exterioare ale tabelului
- `above` - afişează o muchie în partea superioară a cadrului tabelului
- `below` - afişează o muchie în partea inferioară a cadrului tabelului
- `hsides` - afişează câte o muchie în partea superioară şi inferioară cadrului tabelului
- `vsides` - afişează o muchie în partea din stânga şi din dreapta a cadrului tabelului
- `lhs` - afişează o muchie în partea din stânga a cadrului tabelului
- `rhs` - afişează o muchie în partea din dreapta a cadrului tabelului
- `box` - afişează o muchie pe toate laturile cadrului tabelului

Atributul `rules`

Atributul **`rules`** al etichetei <TABLE> permite alegerea unor delimitatori pentru celulele unui tabel. Valorile posibile sunt:

- `none` - elimină toate muchiile interioare ale tabelului
- `rows` - afişează muchii orizontale între toate liniile tabelului
- `cols` - afişează muchii verticale între toate coloanele tabelului
- `all` - afişează muchii între toate liniile şi coloanele

Exemplul 9.21 ilustrează aceste atribute. Atributele `frame` şi `rules` pot fi combinate şi cu atributele de colorare a chenarului şi se pot folosi pentru a crea efecte interesante de aliniere în pagină aşa cum se poate vedea în Figura 9.21.

Exemplul 9. 21

```
<HTML>
  <HEAD>
    <TITLE>tabele21</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Atributele frame si rules</H1><HR>
    <TABLE bordercolor="red" width="400" frame="lhs" rules="cols">
      <TR>
        <TD>celula 11
        <TD>celula 12
        <TD>celula 13
      <TR>
        <TD>celula 21
        <TD>celula 22
        <TD>celula 23
      <TR>
        <TD>celula 21
```

```
        <TD>celula 22
        <TD>celula 23
    </TABLE>
  </BODY>
</HTML>
```

14. Tabele imbricate

Un tabel poate conţine în celulele sale şi alte elemente în afară de text, inclusiv un alt tabel, formând astfel un ansamblu de tabele imbricate.

În Exemplul 9.22 vom ilustra modul în care se pot imbrica două tabele.

Exemplul 9. 22

```
<HTML>
  <HEAD>
    <TITLE>tabele22</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Imbricarea tabelelor</H1><HR>
    <TABLE border="5">
      <TR>
        <TD>celula 11 tabelul 1
        <TD>
          <TABLE border>
            <TR>
              <TD>celula 11 tabelul 2
              <TD>celula 12 tabelul 2
            <TR>
              <TD>celula 21 tabelul 2
              <TD>celula 22 tabelul 2
            <TR>
              <TD>celula 31 tabelul 2
              <TD>celula 32 tabelul 2
          </TABLE>
      <TR>
        <TD>celula 21 tabelul 1
        <TD>celula 22 tabelul 1
    </TABLE>
  </BODY>
</HTML>
```

Aşa cum se vede din Figura 9.22 tabelul este format din două linii şi două coloane. În celula 21 a tabelului am inclus un alt tabel, format din trei linii şi două coloane.

15. Recomandări privind folosirea tabelelor

Tabelele se numără printre cele mai folosite elemente într-o pagină Web. Ele reprezintă un instrument practic indispensabil pentru definirea aspectului paginii, întrucât permit alinierea corectă a elementelor din pagină: text, imagini, legături, etc. În interiorul unui tabel se pot include şi alte tabele, în funcţie de necesităţi, formând o reţea de tabele imbricate.

Singura precauţie pe care trebuie să o aveţi în vedere când folosiţi tabele în paginile dumneavoastră este cea legată de etichetele şi atributele specifice unui anumit tip de browser. Aşa după cum aţi observat din exemplele prezentate, aspectul paginilor poate să difere simţitor de la un tip de browser la altul, atunci când un anumit element nu este recunoscut. Dacă veţi considera că aspectul

tabelelor este esenţial pentru paginile site-ului, este recomandat să vă limitaţi la etichetele standard, recunoscute de toate browserele .
În Anexa 1 veţi găsi o listă a celor mai utilizate etichete împreună cu menţiunea dacă fac parte din standardul HTML 3.2.

16. Rezumat

Tabelele se inserează într-un document HTML cu ajutorul etichetei container `<TABLE> </TABLE>`

Tabelele sunt formate din linii, care se introduc prin eticheta `<TR>` iar fiecare linie este formată din celule introduse prin eticheta `<TD>`.
Celulele unui tabel pot conţine text, etichete de formatare a textului, imagini, legături, formulare sau alte tabele.
Etichetele `<TABLE>`, `<TR>` şi `<TD>` au o serie de atribute comune:

- `align`
  - când este asociat cu `<TABLE>` stabileşte alinierea tabelului faţă de textul din jur
  - când este asociat cu `<TR>` stabileşte alinierea textului din celulele liniei respective
  - când este asociat cu `<TD>` stabileşte alinierea textului din celulă
- `bgcolor` stabileşte culoarea fundalului pentru întregul tabel, o linie sau o celulă.
- `bordercolor`, `bordercolordark` şi `bordercolorlight` stabilesc culorile chenarelor
- `valign` aliniază textul pe verticală în cadrul tabelului, al celulelor unei linii sau al unei celule individuale

Tabelul în întregime sau o celulă a sa pot avea o imagine ca fundal, specificată cu ajutorul atributului `background`.

Grosimea chenarului tabelului se stabileşte cu ajutorul atributului `border`.

Dimensiunile tabelului sau ale unei celule individuale se stabilesc prin intermediul atributelor `width` şi `height`.

Cu ajutorul atributelor `cellspacing` şi `cellpadding` se stabilesc distanţa dintre celulele tabelului, respectiv distanţa dintre conţinutul celulelor şi marginile lor.

Pentru a crea tabele de forme oarecare se folosesc atributele `colspan` şi `rowspan` care permit extinderea unei celule peste celulele din dreapta ei, respectiv peste celulele de sub ea.

Unui tabel i se poate ataşa un titlu prin eticheta `<CAPTION>` şi se pot introduce celule cu rol de cap de tabel prin eticheta `<TH>`.

Într-un tabel se pot defini grupuri de coloane cu ajutorul etichetei container `<COLGROUP> </COLGROUP>`. În cadrul acestui grup, aspectul fiecărei coloane în parte este stabilit prin eticheta `<COL>`.

Modul de afişare al marginilor tabelului precum şi al liniilor despărţitoare dintre linii şi celule se poate stabili cu ajutorul atributelor `frame` şi `rules`.

17. Test

71. *La ce serveşte eticheta* `<TD>`*?*

a) Pentru a insera un tabel în pagină.

b) Pentru a insera o linie într-un tabel.
c) Pentru a insera o celulă de date într-o linie a tabelului.

72. *Următoarea construcţie generează un tabel cu o line şi două celule.*

```
<TABLE bgcolor="white">
  <TR bgcolor="red">
    <TD bgcolor="blue">celula 11
    <TD>Celula 12
</TABLE>
```

Ce culoare vor avea cele două celule?

a) Amândouă alb, deoarece aşa este setată culoarea tabelului.
b) Amândouă roşu, deoarece aşa a fost setată culoarea liniei.
c) Celula 11 albastru, deoarece culoarea ei a fost setată explicit şi celula 12 roşu, deoarece face parte din linia a cărei culoare a fost setată roşu.

73. *La ce servesc atributele* `width` *şi* `height` *ale etichetei* `<TABLE>`*?*

a) Stabilesc dimensiunile tabelului în pagină.
b) Aliniază tabelul faţă de restul elementelor din pagină.
c) Stabilesc dimensiunile celulelor tabelului.

74. *Ce efect are următoarea construcţie:*

```
<TABLE border cellspacing="5" cellpadding="10">?
```

a) Textul din celule este distanţat cu 5 pixeli faţă de chenarul celulelor, iar celulele sunt distanţate între ele cu 10 pixeli.
b) Textul din celule este distanţat cu 10 pixeli faţă de chenarul celulelor, iar celulele sunt distanţate între ele cu 5 pixeli.
c) Chenarul tabelului are grosimea de 5 pixeli iar chenarul celulelor grosimea de 10 pixeli.

75. *Care dintre următoarele construcţii aliniază la dreapta conţinutul celor două celule?*

```
a)  <TR align="left"><TD align="right">celula 11<TD>celula 12
b)  <TR><TD align="right">celula 11<TD>celula 12
c)  <TR align="right"><TD>celula 11<TD>celula 12
```

76. *La ce foloseşte eticheta* `<TH>`*?*

a) Introduce o celulă de date.
b) Introduce o celulă cu rol de cap de tabel.
c) Introduce o linie cu rol de cap de tabel.

77. *Pentru a ataşa un titlu tabelului folosim eticheta:*

a) `<HEAD>`
b) `<CAPTION>`
c) `<TITLE>`

78. *Ce efect are atributul* `colspan` *în construcţia următoare:*

```
<TR><TD colspan="3">cel 11 cel 12 cel 13?
```

a) Unifică trei celule pe orizontală.
b) Unifică trei celule pe verticală.
c) Unifică câte trei celule, atât pe orizontală cât şi pe verticală.

79. *În care dintre construcţiile următoare chenarul tabelului nu este afişat?*

a) `<TABLE border> </TABLE>`
b) `<TABLE border="0"> </TABLE>`
c) `<TABLE border="-1"> </TABLE>`

80. *Atributul* `align` *al etichetei* `<TABLE>` *realizează:*

a) Alinierea tabelului faţă de restul elementelor din pagină.
b) Alinierea textului în tabel.
c) Alinierea textului în fiecare celulă a tabelului.

Tabele

1. c)
2. c)
3. a)
4. b)
5. c)
6. b)
7. b)
8. a)
9. b)
10. a)

Capitolul 10
Cadre (frames)

O modalitate de structurare avansată a unui document HTML este împărţirea ferestrei browserului în mai multe ferestre distincte, denumite cadre (frames). Această facilitate permite afişarea simultană, în aceeaşi fereastră a browserului, a două sau mai multe documente HTML diferite, câte unul în fiecare cadru. Pentru a realiza acest lucru sunt necesare următoarele:

- un document de definire a cadrelor care conţine etichetele care stabilesc numărul, dimensiunile şi aşezarea cadrelor în pagină
- câte un fişier HTML pentru fiecare cadru în parte, prin care se stabileşte conţinutul cadrului respectiv.

De exemplu, o pagină Web care conţine două cadre va fi definită prin trei documente HTML:
- documentul de definire a cadrelor, care specifică dimensiunile şi aşezarea cadrelor în pagină;
- cele două documente care descriu conţinutul fiecărui cadru în parte.

În esenţă, cadrele fac posibilă afişarea în fereastra browserului a mai multe pagini, simultan.

1. Documentul de definire a cadrelor

În cadrul documentului de definire a cadrelor, blocul `<BODY> </BODY>` este înlocuit de blocul **`<FRAMESET> </FRAMESET>`.** În acest tip de document blocul `<BODY>` **nu** mai este folosit.

În interiorul blocului <FRAMESET>, fiecare cadru este introdus prin eticheta **<FRAME>**.

Definirea documentului ce va fi afişat într-un cadru se face prin atributul **src** (*source*). Acesta este un atribut obligatoriu al etichetei <FRAME>, şi primeşte ca valoare adresa URL a documentului HTML care va fi încărcat în acel cadru.

Definirea cadrelor se face prin împărţirea ferestrelor în linii şi coloane:

- împărţirea unei ferestre într-un număr de cadre de tip coloană se face cu ajutorul atributului **cols** al etichetei <FRAMESET> ce descrie acea fereastră;
- împărţirea unei ferestre în cadre de tip linie se face cu ajutorul atributului **rows** al etichetei <FRAMESET> care descrie fereastra.

Valorile atributelor cols si rows sunt liste de elemente separate prin virgulă care descriu modul în care se face împărţirea ferestrei.
Valorile acestor atribute pot fi exprimate în mai multe moduri:

- în pixeli – în care caz valoarea este un număr întreg
- în procente din dimensiunea ferestrei (un număr cuprins între 1 şi 99, urmat de simbolul %)
- în dimensiuni relative, n*. Simbolul n* semnifică faptul că linia sau coloana astfel definită ocupă a *n*-a parte din spaţiul rămas după dispunerea în fereastră a liniilor, respectiv coloanelor precedente (vezi exemplele de mai jos).

Exemplul 1:

```
cols="200 , * , 50% , *
```

Această construcţie descrie o împărţire în patru cadre de tip coloană, dintre care prima are 200 pixeli, a treia ocupă jumătate din spaţiul total disponibil, iar a doua şi a patra ocupă în mod egal restul de spaţiu rămas disponibil.

Exemplul 2:

```
rows="200 , 50% , 1* , 2* "
```

În acest exemplu, pagina este împărţită în patru cadre de tip linie, dintre care prima are 200 pixeli, a doua ocupă jumătate din spaţiul total disponibil iar a treia şi a patra ocupă restul de spaţiu rămas disponibil, care se împarte în trei părţi egale, al treilea cadru ocupând o parte, iar al patrulea ocupând două părţi.

Observaţii

- dacă mai multe elemente din listă sunt configurate cu * , atunci spaţiul disponibil rămas pentru ele se va împărţi în mod egal.
- în cadrul unui bloc <FRAMESET> poate fi inclus un cadru prin eticheta <FRAME> sau un alt bloc <FRAMESET> obţinându-se astfel cadre imbricate.

În majoritatea exemplelor care vor urma am folosit ca URL-uri pentru documentele pe care le conţin cadrele, pagini deja construite în capitolele anterioare. Dacă aţi salvat documentele HTML cu numele specificate în eticheta <TITLE> puteţi verifica direct exemplele care vor urma. Dacă preferaţi să încărcaţi în cadrele descrise în exemplele următoare alte documente, va trebui să faceţi modificările necesare specificând numele acestor fişiere.

Exemplul 10.1 împarte pagina în două cadre verticale, fiecare ocupând jumătate din pagină. Aspectul paginii este cel din Figura 10.1.

Exemplul 10. 1

```
<HTML>
```

```
  <HEAD>
    <TITLE>cadre1</TITLE>
  </HEAD>
  <FRAMESET cols="*, *">
    <FRAME src="liste1.html">
    <FRAME src="liste4.html">
  </FRAMESET>
</HTML>
```

În Exemplul 10.2 este creată o pagina Web cu trei cadre orizontale. Primul cadru are 100 de pixeli, al treilea ocupă 30% din fereastra browserului, iar al doilea ocupă restul spaţiului. Pagina arată ca în Figura 10.2.

Exemplul 10. 2

```
<HTML>
  <HEAD>
    <TITLE>cadre2</TITLE>
  </HEAD>
  <FRAMESET rows="100 , * , 30%">
    <FRAME src="liste1.html">
    <FRAME src="liste2.html">
    <FRAME src="liste3.html">
  </FRAMESET>
</HTML>
```

Exemplul 10.3 crează o matrice pătrată de 4 cadre (2 x 2). Pentru a realiza acest lucru, se folosesc simultan cele două atribute `cols` şi `rows.` Pagina descrisă în acest document va avea aspectul din Figura 10.3.

Exemplul 10. 3

```
<HTML>
  <HEAD>
    <TITLE>cadre3</TITLE>
  </HEAD>
  <FRAMESET rows="*, *" cols="*, *">
    <FRAME src="tabele1.html">
    <FRAME src="tabele2.html">
    <FRAME src="tabele4.html">
    <FRAME src="tabele6.html">
  </FRAMESET>
</HTML>
```

2. Cadre imbricate

Cadrele, ca şi alte elemente ale paginii Web (liste, tabele), se pot imbrica, adică pot fi incluse cadre în interiorul altor cadre. În Exemplul 10.4 este creată o pagină cu trei cadre mixte. Pentru a construi pagina se procedează din aproape în aproape. Mai întâi, pagina este împărţită în două cadre de tip coloană, după care al doilea cadru este împărţit în două cadre de tip linie. Pagina va avea aspectul din Figura 10.4.

Exemplul 10. 4

```
<HTML>
  <HEAD>
    <TITLE>cadre4</TITLE>
  </HEAD>
  <FRAMESET cols="30%, *">
    <FRAME src="tabele1.html">
    <FRAMESET rows="*, *">
      <FRAME src="tabele2.html">
      <FRAME src="tabele4.html">
    </FRAMESET>
  </FRAMESET>
```

```
</HTML>
```

3. Controlul aspectului unui cadru

Culoarea marginii cadrului

Pentru a stabili culoarea chenarului unui cadru se utilizează atributul **bordercolor**. Acest atribut primeşte ca valoare un nume de culoare sau o culoare definită în conformitate cu modelul RGB.

Atributul `bordercolor` poate fi ataşat atât etichetei <FRAMESET> pentru a stabili culoarea tuturor chenarelor cadrelor incluse, cât şi etichetei <FRAME>, pentru a stabili culoarea chenarului pentru un cadru individual.

Exemplul 10.5 ilustrează atributul `bordercolor`. Pagina descrisă în acest document are aspectul din Figura 10.5.

Exemplul 10. 5

```
<HTML>
  <HEAD>
    <TITLE>cadre5</TITLE>
  </HEAD>
  <FRAMESET cols="20%, *" bordercolor="green" border="15">
    <FRAME src="text3.html">
    <FRAMESET rows="*, *">
      <FRAME src="text4.html" bordercolor="blue">
      <FRAME src="text5.html">
    </FRAMESET>
  </FRAMESET>
</HTML>
```

Lăţimea marginii cadrului

Atributul **border** al etichetei <FRAMESET> permite configurarea lăţimii chenarelor tuturor cadrelor la un număr dorit de pixeli. Valoarea atributului `border` este un număr întreg, ce reprezintă numărul de pixeli, valoarea prestabilită fiind de 5 pixeli. În mod prestabilit, chenarul unui cadru este afişat şi are aspect tridimensional.
Pentru a obţine cadre fără chenar se utilizează setarea `border="0"`.

Afişarea chenarului unui cadru se mai poate dezactiva şi dacă se utilizează atributul **frameborder** cu valoarea "no". Acest atribut poate fi ataşat atât etichetei <FRAMESET> (dezactivarea fiind valabilă pentru toate cadrele incluse) cât şi etichetei <FRAME> (dezactivarea fiind valabila numai pentru un singur cadru).
Valorile posibile ale atributului `frameborder` sunt:

- `yes` - echivalent cu 1
- `no` - echivalent cu 0

Cadrele din Exemplul 10.6 nu au chenar. După cum puteţi observa din Figura 10.6 crearea unor cadre fără chenar poate duce la apariţia unor ambiguităţi în pagină aşa încât acest efect trebuie folosit cu atenţie.

Exemplul 10. 6

```
<HTML>
  <HEAD>
    <TITLE>cadre6</TITLE>
  </HEAD>
  <FRAMESET rows="*, *" border="0">
    <FRAME src="tabele1.html">
    <FRAME src="tabele2.html">
  </FRAMESET>
```

```
</HTML>
```

Dimensiunea cadrului

Chiar dacă dimensiunile unui cadru au fost stabilite în mod explicit prin valorile atributelor etichetei <FRAMESET>, utilizatorul are posibilitatea de a altera aceste dimensiuni cu ajutorul mouse-ului.
În scopul prevenirii acestei situaţii se poate utiliza atributul **noresize**, ataşat etichetei <FRAME>, al cărui efect este cel de blocare a posibilităţii de redimensionare a cadrului.

4. Bare de derulare

Atributul **scrolling** al etichetei <FRAME> este utilizat pentru a adăuga unui cadru o bară de derulare sau de defilare (*scrolling bar*), care permite navigarea în interiorul documentului afişat în cadru. Valorile posibile ale atributului `scrolling` sunt:

- `yes` - bara de derulare este prezentă întotdeauna;
- `no` - bara de derulare nu este disponibilă;
- `auto` - bara de derulare este vizibilă atunci când este necesar. Aceasta opţiune lasă browserului posibilitatea de a adăuga sau nu bara, în funcţie de dimensiunea textului din cadru.

Exemplul 10.7 ilustrează funcţionarea atributului `scrolling` în cele trei situaţii.

Exemplul 10. 7

```
<HTML>
  <HEAD>
    <TITLE>cadre7</TITLE>
  </HEAD>
  <FRAMESET rows="*, *, *">
    <FRAME src="text1.html" scrolling="yes" noresize>
    <FRAME src="text7.html" scrolling="no" noresize>
    <FRAME src="text8.html" scrolling="auto" noresize>
  </FRAMESET>
</HTML>
```

Din Figura 10.7 se observă că, deoarece la primul cadru textul nu depăşeşte dimensiunea ferestrei, bara de derulare nu este afişată deşi atributul `scrolling` are valoarea "yes".
În cel de-al doilea cadru bara de derulare nu este afişată, indiferent de dimensiunea textului din cadru.
La cel de-al treilea cadru, prezenţa sau absenţa barei de derulare este condiţionată de dimensiunea textului. Aici, deoarece textul depăşeşte dimensiunea ferestrei, bara este afişata.

5. Poziţionarea documentului într-un cadru

Atributele **marginheight** şi **marginwidth** ale etichetei <FRAME> permit stabilirea distanţei în pixeli dintre conţinutul unui cadru şi marginile verticale, respectiv orizontale ale cadrului. Valorile posibile ale acestor atribute sunt:

- un număr de pixeli
- procente din înălţimea, respectiv din lăţimea cadrului

În Exemplul 10.8 fereastra browserului este împărţită în trei cadre de tip coloană de dimensiuni egale. În toate cele trei cadre este afişat acelaşi document, ceea ce diferă este poziţionarea documentului în fiecare cadru.
Astfel, în cadrul din stânga, nu este specificată nici o valoare pentru atributele `marginheight` şi `marginwidth`.

În cadrul din mijloc, se stabileşte distanţa de 50 de pixeli între marginea superioară şi cea inferioară a cadrului şi text.
În cadrul din dreapta se stabileşte o distanţă de 30 de pixeli între marginea din stânga şi cea din dreapta a cadrului şi text. Pagina are aspectul din Figura 10.8.

Exemplul 10. 8

```
<HTML>
  <HEAD>
    <TITLE>cadre8</TITLE>
  </HEAD>
  <FRAMESET cols="* , * , *">
    <FRAME src="text1.html">
    <FRAME src="text1.html" marginheight="60">
    <FRAME src="text1.html" marginwidth="40">
  </FRAMESET>
</HTML>
```

Atenţie!

Există browsere care nu suportă cadre. Pentru acestea se utilizează în interiorul blocului <FRAMESET> eticheta **<NOFRAMES> </NOFRAMES>**. Dacă browserul poate să interpreteze cadre, va ignora ce se găseşte în această porţiune, iar dacă nu, materialul cuprins în zona <NOFRAMES> </NOFRAMES> va fi singurul care va fi recunoscut şi afişat.

Este de precizat şi faptul că între <NOFRAMES> </NOFRAMES> se pot introduce orice alte etichete HTML (inclusiv imagini, hyperlink-uri, tabele).

6. Cadre interne (in-line frames)

Un cadru intern este specificat prin intermediul etichetei **<IFRAME> </IFRAME>**. Ea defineşte o arie rectangulară în interiorul documentului, arie în care browserul va afişa un alt document HTML, complet, inclusiv marginile şi barele de derulare.
Un cadru intern se inserează într-o pagina Web în mod asemănător cu o imagine, în interiorul blocului <BODY>, aşa cum rezultă din următorul exemplu:

```
<IFRAME src="tabele10.html" height=40% width=50%> </IFRAME>
```

În acest caz, am specificat că dorim un cadru intern care are 40% din înălţimea şi 50% din lăţimea paginii curente.

Atributele acceptate de eticheta <IFRAME> sunt în parte preluate de la etichetele <FRAME> şi <FRAMESET>, cum ar fi:
src, border, frameborder, bordercolor, marginheight, marginwidth, scrolling, name.
O parte din atributele <FRAME> sunt comune cu cele ale etichetei <IMG>:
vspace, hspace, align, width, height.

În Exemplul 10.9 am construit un cadru intern în care se deschide unul dintre documentele HTML create la capitolul **Tabele**.

Exemplul 10. 9

```
<HTML>
  <HEAD>
    <TITLE>cadre9</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Cadre interne</H1><HR>
    <CENTER>
      <IFRAME width="60%" height="50%" name="icad" src="tabele4.html">
      Daca vedeti acest text inseamna ca browserul dumnevoastra nu suporta
      cadre interne.
      <P><A href="cadre10.html">Pagina fara cadre interne</A>
```

```
      </IFRAME>
    </CENTER>
  </BODY>
</HTML>
```

Din Figura 10.9 puteţi observa că aspectul unui cadru intern este întrucâtva similar cu cel al unei imagini *in-line*.
Eticheta <IFRAME> se introduce în cadrul blocului <BODY> şi în cazul folosirii ei nu mai este necesar un document de definire a cadrelor. Pentru situaţia când browserul nu acceptă cadre interne, am asigurat o versiune a paginii care nu conţine astfel de cadre (*cadre10.html*) introdusă prin eticheta <A>. Conţinutul acestei pagini este foarte simplu, servind doar la ilustrarea modului în care poate fi construită o alternativă la pagina cu cadre interne (Exemplul 10.10)

Exemplul 10. 10

```
<HTML>
  <HEAD>
    <TITLE>cadre10</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Pagina fara cadre interne</H1><HR>
  </BODY>
</HTML>
```

Iată un alt exemplu (Exemplul 10.11) în care am creat o pagină care conţine trei link-uri iar acestea deschid paginile referite de ele în cadrul intern din centrul paginii. Figura 10.10 redă aspectul acestei pagini.

Exemplul 10. 11

```
<HTML>
  <HEAD>
    <TITLE>cadre11</TITLE>
  </HEAD>
  <BODY>
    <A href="tabele4.html" target="icad">Fisierul1</A><BR>
    <A href="tabele5.html" target="icad">Fisierul2</A><BR>
    <A href="tabele6.html" target="icad">Fisierul3</A><BR>
    <CENTER>
      <IFRAME width="60%" height="50%" name="icad" src="cadre11.html">
      Daca vedeti acest text inseamna ca browserul dumnevoastra nu suporta
      cadre interne.
      <P><A href="cadre10.html">Pagina fara cadre interne</A>
      </IFRAME>
    </CENTER>
  </BODY>
</HTML>
```

În exemplu am folosit atributul **name** al etichetei <IFRAME> cu ajutorul căruia am atribuit un nume acestui cadru şi anume "icad". Acest lucru a fost necesar pentru a specifica, prin intermediul atributului **target** al etichetei <A>, faptul că link-urile se vor deschide în cadrul intern.

7. Deschiderea documentelor în alte cadre

Dacă într-unul dintre documentele deschise în cadru există link-uri, acestea vor deschide paginile referite de ele în cadrul curent.

Acest comportament se poate schimba prin plasarea în eticheta <A> a atributului **target**, care precizează numele ferestrei (cadrului) în care se va încărca pagina nouă referită de legătură, conform sintaxei:

```
<A href="URL" target="nume_frame"> </A>
```

Numele unui cadru este stabilit prin atributul **name** al etichetei <FRAME> conform sintaxei:

```
<FRAME name="nume_frame">
```

În Exemplul 10.12 este prezentată o pagina Web cu două cadre. Documentul de definire a cadrelor este *cadre12. html* al cărui aspect este redat în Figura 10.11.

Exemplul 10. 12

```
<HTML>
  <HEAD>
    <TITLE>cadre12</TITLE>
  </HEAD>
  <FRAMESET cols="20%, *">
    <FRAME src="cadre13.html">
    <FRAME src="cadre14.html" name="cadru_dreapta">
  </FRAMESET>
</HTML>
```

Documentul de definire a cadrelor împarte pagina în două cadre de tip coloană. În cel din stânga se va deschide documentul *cadre13.html* iar în cel din dreapta, documentul *cadre14.html*. Cel de-al doilea cadru a fost numit "cadru_dreapta".

În Exemplul 10.13 este creat documentul *cadre13.html*. Acest document conţine, după cum observaţi, patru link-uri. Prin intermediul atributului target am specificat cadrul în care se vor deschide acestea. Toate aceste legături deschid paginile referite de ele în cadrul drept.

Exemplul 10. 13

```
<HTML>
  <HEAD>
    <TITLE>cadre13</TITLE>
  </HEAD>
  <BODY>
    <A href="tabele6.html" target="cadru_dreapta">
      Legatura 1
    </A><BR>
    <A href="tabele7.html" target="cadru_dreapta">
      Legatura 2
    </A><BR>
    <A href="tabele8.html" target="cadru_dreapta">
      Legatura 3
    </A><BR>
    <A href="cadre14.html" target="cadru_dreapta">
      Home
    </A><BR>
  </BODY>
</HTML>
```

În Exemplul 10.14 este creat documentul *cadre14.html*.

Exemplul 10. 14

```
<HTML>
  <HEAD>
    <TITLE>cadre14</TITLE>
  </HEAD>
  <BODY>
    <H3>Aici se vor deschide paginile referite de legaturile din cadrul
    stang</H3><HR>
  </BODY>
</HTML>
```

Salvaţi cele trei fişiere cu numele specificate şi deschideţi cu browserul fişierul *cadre12.html* pentru a observa cum funcţionează atributele prezentate.

Atributul `target` al etichetei `<FRAME>` acceptă şi anumite valori predefinite şi anume:

- `_self` - încărcarea noii pagini are loc în cadrul curent
- `_blank` - încărcarea noii pagini are loc într-o fereastră nouă, anonimă
- `_parent` - încărcarea noii pagini are loc în cadrul părinte al cadrului curent dacă acesta există, altfel are loc în fereastra curentă a browserului
- `_top` - încărcarea noii pagini are loc în fereastra browserului ce conţine cadrul curent.

Următorul exemplu ilustrează felul cum funcţionează aceste valori ale atributului `target`.
Documentul de definire a cadrelor este prezentat în Exemplul 10.15 aspectul său fiind cel din Figura 10.12. Pagina construită în acest exemplu conţine două cadre orizontale. În cel de sus este încărcat documentul *cadre17.html* iar în cel de jos, documentul *cadre16.html*.

Exemplul 10. 15

```
<HTML>
  <HEAD>
    <TITLE>cadre15</TITLE>
  </HEAD>
  <FRAMESET rows="30%, *">
    <FRAME src="cadre17.html">
    <FRAME src="cadre16.html">
  </FRAMESET>
</HTML>
```

Exemplul 10.16 construieşte documentul *cadre16.html* care conţine trei legături către fişiere HTML create la capitolul **Formatarea textului**. Fiecare dintre aceste legaturi are specificat atributul `target` pentru a ilustra modul în care funcţionează acesta.

Exemplul 10. 16

```
<HTML>
  <HEAD>
    <TITLE>cadre16</TITLE>
  </HEAD>
  <BODY>
    <A href="text4.html" target="_blank">
      Aceasta legatura se deschide in alta fereastra
    </A><BR>
    <A href="text5.html" target="_self">
      Aceasta legatura se deschide in fereastra curenta
    </A><BR>
    <A href="text6.html" target="_parent">
      Aceasta legatura se deschide in fereastra parinte
    </A><BR>
</BODY>
</HTML>
```

Documentul *cadre17.html* prezentat în Exemplul 10.17 este o pagină Web foarte simplă care se va deschide în cadrul de sus.

Exemplul 10. 17

```
<HTML>
  <HEAD>
    <TITLE>cadre17</TITLE>
  </HEAD>
  <BODY>
```

```
    <H1 align="center">Atributul target</H1>
  </BODY>
</HTML>
```

Pentru a observa modul de funcţionare al atributului `target` salvaţi cele trei fişiere sub numele specificate şi deschideţi cu browserul documentul *cadre15.html*.

8. Sugestii privind folosirea cadrelor

Deşi cadrele reprezintă o maniera destul de spectaculoasă de a structura o pagină Web, ele prezintă o mulţime de dezavantaje.
Încărcarea unei pagini care conţine cadre se face mai greu, indexarea paginii de către motoarele de căutare este mai dificilă. De asemenea, deoarece documentul din fiecare cadru are propriul URL , este mai greu pentru vizitator să reţină adresa paginii în *Favorites*. Şi, nu în ultimul rând, deoarece există browsere care nu suportă cadrele, este indicat să asiguraţi pentru fiecare pagină astfel structurată şi o versiune fără cadre, ceea ce implică un efort suplimentar.

Având în vedere toate aceste aspecte, este bine să limitaţi folosirea cadrelor în paginile dumneavoastră doar la situaţiile în care nu puteţi proceda altfel.

9. Rezumat

Folosirea cadrelor într-un document HTML permite deschiderea mai multor pagini distincte în documentul respectiv.

Pentru a crea o pagină care conţine cadre sunt necesare mai multe documente HTML:
- documentul de definire a cadrelor
- câte un document HTML pentru fiecare cadru definit

Documentul de definire a cadrelor nu conţine blocul `<BODY>` acesta fiind înlocuit de blocul `<FRAMESET> </FRAMESET>`. În cadrul blocului `<FRAMESET>` fiecare cadru este introdus prin eticheta `<FRAME>`.

Împărţirea ferestrei în cadre de tip coloană se face cu ajutorul atributului `cols` iar în cadre de tip linie se face cu ajutorul atributului `rows`. Ambele atribute sunt asociate etichetei `<FRAMESET>`.

Pentru a stabili lăţimea chenarului unui cadru se foloseşte atributul `border` al etichetei `<FRAME>`. Pentru a stabili culoarea chenarului unui cadru se foloseşte atributul `bordercolor` al etichetei `<FRAME>`. Aceste atribute pot fi asociate şi etichetei `<FRAMESET>`, caz în care efectul lor se va aplica tuturor cadrelor din set.

Prezenţa barelor de derulare într-un cadru se stabileşte cu ajutorul atributului `scrolling`.

Pentru a poziţiona documentul într-un cadru se folosesc atributele `marginwidth` şi `marginheight`.

Într-o pagină Web se pot introduce şi cadre interne, cu ajutorul etichetei `<IFRAME> </IFRAME>`.

Pentru a specifica fereastra sau cadrul în care se va deschide un document se foloseşte atributul `target` care poate avea ca valoare numele ferestrei (cadrului) sau poate avea o valoare predefinită.

Este recomandat să asiguraţi o versiune fără cadre a documentului HTML, pentru ca acesta să poată fi vizualizat cu browserele care nu suportă cadre. Pentru aceasta, elementele care fac parte din versiunea fără cadre se includ între etichetele `<NOFRAMES> </NOFRAMES>`.

10. Test

81. *În ce zonă a unui document HTML se introduce blocul* `<FRAMESET>`*?*

a) În blocul `<BODY>`
b) În blocul `<HEAD>`
c) `<FRAMESET>` formează un bloc separat.

82. *Cum se introduce un cadru în pagină?*

a) Prin eticheta `<FRAMESET>`
b) Prin eticheta `<FRAME>`
c) Prin eticheta `<SRC>`

83. *Dorim să împărţim o pagină în trei cadre de tip coloană, primul având jumătate din pagină, iar celelalte două spaţiul rămas împărţit în mod egal. Care dintre construcţiile următoare realizează acest lucru?*

a) `<FRAMESET cols="* , *, *"`
b) `<FRAMESET cols="50%, 1* , 2*"`
c) `<FRAMESET cols="50%, * , *"`

84. *Ce realizează următoarea construcţie?*

```
<HTML>
  <FRAMESET cols="*, *">
    <FRAME src="frame1.html">
    <FRAMESET rows="*, *">
      <FRAME src="frame2.html">
      <FRAME src="frame3.html'>
    </FRAMESET>
  </FRAMESET>
</HTML>
```

a) Împarte pagina în patru cadre, două de tip coloană, două de tip linie.
b) Împarte pagina în două cadre de tip coloană, cel de-al doilea fiind împărţit în două cadre de tip linie.
c) Împarte pagina în trei cadre de tip linie.

85. *În construcţia următoare la ce foloseşte atributul* `scrolling`*?*

```
<FRAME src="frame1.html" scrolling="auto">
```

a) Permite auto-dimensionarea cadrului în funcţie de mărimea textului.
b) Adaugă automat bara de derulare pentru acest cadru.
c) Lasă browserului posibilitatea de a adăuga sau nu bara de derulare, în funcţie de mărimea textului din cadru.

86. *La ce folosesc atributele* `marginheight` *şi* `marginwidth` *ale etichetei* `<FRAME>`*?*

a) Dimensionează marginile cadrului.
b) Dimensionează chenarul cadrului.
c) Stabilesc distanţa dintre marginile cadrului şi conţinutul său.

87. *Unde se introduce eticheta* `<IFRAME>`*?*

a) În blocul `<BODY>`
b) În blocul `<FRAMESET>`
c) În blocul `<FRAME>`

88. *La ce serveşte atributul* `bordercolor` *asociat etichetei* `<FRAMESET>`*?*

a) Stabileşte culoarea chenarului unui cadru.
b) Stabileşte culoarea chenarelor tuturor cadrelor definite în <FRAMESET>.
c) Stabileşte culoarea de fond a cadrelor definite în <FRAMESET>.

89. *Care dintre următoarele construcţii este corectă?*

a) <FRAME name="nume_cadru">
b) <FRAMESET name="nume_cadru">
c) <FRAMESET href="nume_cadru">

90. *Cum se realizează deschiderea documentului dintr-un cadru într-o fereastra nouă?*

a) <FRAME src="frame1.html" target="_self">
b) <FRAME src="frame1.html" target="_blank">
c) <FRAME src="frame1.html" target="_parent">

Cadre

1. c)
2. b)
3. c)
4. b)
5. c)
6. c)
7. a)
8. b)
9. a)
10. b)

Capitolul 11
Formulare

Cu siguranţă aţi văzut cum arată un formular într-o pagină Web. De exemplu, atunci când completaţi o carte de oaspeţi (*guestbook*) aveţi de-a face cu un formular. Atunci când selectaţi mai multe opţiuni dintr-o listă sau introduceţi un cuvânt într-un motor de căutare, folosiţi, de asemenea, un formular.

Formularul reprezintă unul dintre cele puternice elemente ale unei pagini Web. Prin intermediul formularelor se realizează interactivitatea cu vizitatorul paginii, acestea permiţându-vă să obţineţi informaţii despre cei care vizitează paginile dumneavoastră.

1. Ce este un formular?

Un formular este un ansamblu de zone active alcătuit din mai multe tipuri de elemente: butoane, casete de selecţie, câmpuri de editare, etc., ce permit utilizatorilor să introducă efectiv informaţii. Aceste informaţii sunt ulterior transmise serverului pe care este găzduită pagina dumneavoastră, unde vor putea fi prelucrate.

O sesiune cu o pagina Web ce conţine un formular cuprinde două etape:

- Utilizatorul completează formularul şi, prin apăsarea butonului de expediere, trimite serverului datele înscrise în formular.
- O aplicaţie dedicată de pe server (un script) analizează informaţiile transmise şi, în funcţie de configuraţia scriptului, fie stochează datele într-o bază de date, fie le transmite la o adresă de mail indicată de dumneavoastră. Dacă este necesar, serverul poate expedia şi un mesaj de răspuns utilizatorului.

Cel mai important lucru pe care trebuie să-l înţelegeţi în legătură cu formularele este că aici apar două probleme distincte şi care necesită instrumente diferite pentru a le gestiona.

Prima dintre ele este plasarea formularului în pagină şi asigurarea bunei lui funcţionări. Acest lucru se realizează prin HTML şi de el ne ocupăm în acest capitol.

Cea de-a doua este gestionarea şi prelucrarea informaţiilor pe care vizitatorul le introduce prin intermediul formularului. Această problemă este rezolvată cu ajutorul scripturilor CGI stocate pe server, subiect despre care vom vorbi în capitolul următor.

Structura formularele poate varia, de la o simplă casetă de text pentru introducerea unui şir de caractere, până la un ansamblu complex, cu multiple secţiuni şi care oferă facilităţi puternice de transmitere/prelucrare a datelor.

2. Crearea unui formular

Un formular este definit într-un bloc delimitat de etichetele **<FORM> </FORM>.** Eticheta `</FORM>`, de închidere, este obligatorie.
În interiorul blocului sunt incluse:

- elementele formularului, în care vizitatorul urmează să introducă informaţii
- un buton de expediere (*submit*), la apăsarea căruia, datele sunt transmise către server
- opţional, un buton de anulare (*reset, cancel*), prin care utilizatorul poate anula datele înscrise în formular

Tot ceea ce este inclus între eticheta de deschidere şi cea de închidere face parte din formular.

Eticheta `<FORM>` are două atribute esenţiale:

`action` – comunică browserului unde să trimită datele introduse în formular. În general valoarea atributului `action` este adresa URL a scriptului aflat pe serverul care primeşte datele formularului. De exemplu:

```
<FORM action="http://www.yahoo.com/cgi-bin/nume_fis.cgi">.
```

`method` - precizează metoda utilizată de browser pentru expedierea datelor formularului.

Atributul `method` poate avea două valori:

- `post` - folosită în cele mai multe cazuri. Indică serverului să furnizeze datele direct scriptului ca date de intrare standard.

- `get` (valoarea implicită) - datele din formular sunt adăugate la adresa URL precizată de atributul `action`, între adresa URL şi date fiind inserat un "?". Datele sunt adăugate conform sintaxei:
  `nume_camp=valoare_camp.`
  Între diferite seturi de date este introdus un "&". De exemplu:

  ```
  "http://www.yahoo.com/cgi-bin/nume_fis.cgi?
  nume1=valoare1&nume2=valoare2"
  ```

Cea mai facilă cale prin care informaţiile introduse într-un formular pot parveni creatorului paginii este folosirea comenzii **`mailto:`** Am mai întâlnit această comandă la capitolul **Legături** în contextul următor:

```
<A href="mailto:autor@domeniu.com">
```

De data aceasta nu mai este vorba de crearea unei legături care să lanseze în execuţie aplicaţia de mail a utilizatorului, ci de transmiterea datelor introduse în formular la o adresă de mail specificată de dumneavoastră.

Acest lucru se poate realiza precizând ca valoare a atributului `action` următoarea secvenţă: `"mailto:adresa_mail",` ca în exemplul următor:

```
<FORM action=mailto:autor@domeniu.com method="post">
```

Aşa cum spuneam, atributul `action` comunică browserului unde să trimită datele introduse în formular. Folosind comanda `mailto:` browserului i se indică să trimită datele la adresa de mail specificată în comandă.

Din păcate, această comandă nu este executată în acest mod decât de browserele Netscape. Browserele Internet Explorer nu recunosc comanda `mailto:` prezentă în cadrul formularelor. Acestea o execută la fel cum este ea executată atunci când face parte din eticheta `<A>` ca valoare a atributului `href` adică lansează aplicaţia de mail a utilizatorului.

Din acest motiv, inclusiv pentru trimiterea datelor la o adresă specificată de mail, se folosesc scripturi CGI. Pentru a nu intra încă în amănunte privind scripturile, vom folosi totuşi în exemplele următoare comanda `mailto:`
Reţineţi totuşi că afişarea corectă a unui formular în pagină nu este decât prima etapă a utilizării formularului. Pentru a face o testare completă trebuie să aveţi un script instalat pe server care să prelucreze datele din formular şi să le transmită la adresa de mail specificată.

Pentru a defini elementele care fac parte din formular se utilizează etichetele **`<INPUT>`**, **`<SELECT>`**, **`<OPTION>`** şi **`<TEXTAREA>`**.

3. Eticheta `<INPUT>`

Prin intermediul etichetei **`<INPUT>`** se pot introduce în formular câmpuri de editare (casete de text), câmpuri de tip *password*, butoane de expediere şi anulare, butoane radio şi casete de validare.

Eticheta `<INPUT>` are următoarele atribute:

`type` - prin care se precizează tipul elementului.
`name` - permite ataşarea unui nume fiecărui element al formularului.
`value` - permite atribuirea unei valori iniţiale unui element al formularului.

Perechea de atribute **name/value** (nume/valoare) este deosebit de importantă pentru buna funcţionare a formularului.

Fiecare element introdus prin eticheta <INPUT> reprezintă o variabilă. Informaţia introdusă de utilizator în câmpul elementului respectiv reprezintă valoarea pe care o primeşte această variabilă. Din acest motiv, toate elementele introduse prin eticheta <INPUT> trebuie să aibă atribuit un nume. Acest lucru este obligatoriu deoarece, în caz contrar, variabila reprezentată de acel element nu va avea un identificator.
În plus, este indicat să atribuiţi nume distincte diferitelor elemente care fac parte din formular. Altminteri, scriptul CGI care prelucrează datele nu va şti să facă distincţia între variabile având acelaşi nume şi valori diferite.

Atributul type poate avea următoarele valori:

- text - se foloseşte pentru a introduce un câmp de editare pe o singură linie
- password – se foloseşte pentru a insera un câmp de editare în care caracterele introduse sunt înlocuite cu asteriscuri (*)
- radio – folosit pentru a insera un şir de butoane radio (elemente care se folosesc pentru a selecta dintr-o listă de opţiuni una şi numai una dintre ele)
- checkbox – folosit pentru a introduce un şir de casete de validare (elemente care se folosesc când dintr-o listă de opţiuni se poate alege mai mult de o singură variantă)
- submit – se foloseşte pentru a introduce un buton de expediere
- reset – se foloseşte pentru a introduce un buton de anulare a informaţiilor introduse în formular
- button - se foloseşte pentru a introduce în formular un buton obişnuit
- hidden - se foloseşte pentru a introduce în formular un câmp ascuns

Fiecare dintre valorile atributului type generează un anumit tip de element în cadrul formularului.

Exemplul 11.1 creează un formular foarte simplu care conţine un câmp de editare şi un buton de expediere. Aspectul formularului este redat în Figura 11.1.

Exemplul 11. 1

```
<HTML>
  <HEAD>
    <TITLE>formulare1</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center"> Caseta de text si buton de expediere</H1><HR>
    <FORM action="mailto:adresa@yyy.com" method="post">
      Introduceti adresa dvs. de mail:
      <INPUT type="text" name="adresa" value="adresa mail"><BR>
      <INPUT type="submit" value="expediaza">
    </FORM>
  </BODY>
</HTML>
```

După cum observaţi, numele atribuit câmpului de editare este "adresa" iar atributul value a primit ca valoare şirul de caractere "adresa mail". Acest text va fi afişat în interiorul câmpului de editare, înainte ca utilizatorul să înceapă introducerea informaţiilor. Dacă atributul value nu era prezent sau era iniţializat cu stringul vid (""), caseta de text ar fi fost goală.

În cadrul formularului este prezent şi un buton de expediere, inserat de asemenea prin intermediul etichetei <INPUT> având atributul type setat cu

valoarea "submit". Aşa cum se observă din figură, pe buton este afişat textul "expediaza", deoarece pentru acest element, atributului value i-a fost atribuită ca valoare textul respectiv.

Casete de text

În Exemplul 11.2 am creat un formular care conţine trei casete de text (câmpuri de editare) în care utilizatorul este rugat să introducă numele, prenumele şi adresa de mail, precum şi o casetă de tip *password*, care cere introducerea unei parole. Aşa cum veţi observa din Figura 11.2, o casetă de tip *password* este asemănătoare cu un câmp de editare, singura diferenţă fiind aceea că aici caracterele nu sunt afişate în clar ci sunt înlocuite cu asteriscuri.
În plus, formularul conţine un buton de expediere şi unul de anulare a datelor introduse.

Exemplul 11. 2

```
<HTML>
  <HEAD>
    <TITLE>formulare2</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Casete de text</H1><HR>
    <FORM action="mailto:adresa@yyy.com" method="post">
      Introduceti numele:
      <INPUT type="text" name="nume" value=""><BR>
      Introduceti prenumele:
      <INPUT type="text" name="prenume" value=""><BR>
      Introduceti adresa de mail:
      <INPUT type="text" name="mail" value=""><BR>
      Introduceti parola:
      <INPUT type="password" name="parola"><BR>
      <INPUT type="reset" value="sterge">
      <INPUT type="submit" value="expediaza">
    </FORM>
  </BODY>
</HTML>
```

Pentru elementele de tip casetă de text şi *password* sunt utile şi atributele:

maxlength – care stabileşte numărul maxim de caractere care poate fi introdus în câmpul de editare
size – care stabileşte lăţimea acestui câmp

Butoane radio

În Exemplul 11.3 am creat un formular care conţine un şir de butoane radio, prin care se cere părerea vizitatorului despre o pagină Web. După cum observaţi, fiecare element de tip buton radio are acelaşi nume şi anume "opinie" deoarece nu se poate selecta decât un singur element, astfel încât orice ambiguitate este exclusă.
Fiecare buton radio trebuie să aibă o valoare. Din acest motiv pentru fiecare buton atributul value a fost setat la o altă valoare. Alegerea uneia dintre opţiuni, de exemplu "buna", face ca variabila "opinie" să primească valoarea "buna". Prin urmare, serverului îi va fi transmisă perechea "opinie=buna".
În plus, observaţi prezenţa atributului **checked**, care are rolul de a prestabili o anumită opţiune, pe care însă utilizatorul o poate schimba, dacă doreşte.
Aspectul formularului este cel din Figura 11.3.

Exemplul 11. 3

```
<HTML>
  <HEAD>
    <TITLE>formulare3</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Butoane radio</H1><HR>
    <FORM action="mailto:adresa@yyy.com" method="post">
      Ce parere aveti despre aceasta pagina?<BR>
      <INPUT type="radio" name="opinie" value="foarte buna" checked>Foarte
      buna<BR>
      <INPUT type="radio" name="opinie" value="buna">Buna<BR>
      <INPUT type="radio" name="opinie" value="destul de buna">Destul de
      buna<BR>
      <INPUT type="radio" name="opinie" value="proasta">Proasta<BR>
      <INPUT type="reset" value="anuleaza">
      <INPUT type="submit" value="expediaza">
    </FORM>
  </BODY>
</HTML>
```

Casete de validare

Exemplul 11.4 creează un şir de casete de validare, care se deosebesc de butoanele radio prin faptul că se pot selecta mai multe opţiuni dintre cele prezentate. Figura 11.4 redă aspectul formularului construit în acest exemplu.

Exemplul 11. 4

```
<HTML>
  <HEAD>
    <TITLE>formulare4</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Casete de validare</H1><HR>
    <FORM action="mailto:adresa@yyy.com" method="post">
      Care sunt hobby-urile dumneavoastra?<BR>
      <INPUT type="checkbox" name="hobby" value="sport">Sport<BR>
      <INPUT type="checkbox" name="hobby" value="film">Film<BR>
      <INPUT type="checkbox" name="hobby" value="lectura">Lectura<BR>
      <INPUT type="checkbox" name="hobby" value="jocuri">Jocuri pe
      computer<BR>
      <INPUT type="checkbox" name="hobby" value="internet">Surfing pe
      net<BR>
      <INPUT type="checkbox" name="hobby" value="alpinism">Alpinism<BR>
      <INPUT type="submit" name="submit" value="expediaza">
    </FORM>
  </BODY>
</HTML>
```

O casetă de validare are două stări: marcată sau nemarcată (*on* sau *off*). Ca şi la butoanele radio, atributul `name` are în exemplul considerat o singură valoare, `"hobby"`, iar atributul `value` are, pentru fiecare casetă, valori distincte. Serverului îi sunt transmise numai acele valori care au fost marcate în casetele de validare corespunzătoare.

Şi în cazul casetelor de validare este posibilă prezenţa atributului **`checked`** care preselectează o anumită opţiune sau mai multe.

Butoane

În cadrul unui formular se pot introduce şi butoane obişnuite (altele decât butoanele *Submit* şi *Reset*). Rolul acestora este de a iniţia anumite acţiuni în momentul când utilizatorul face click cu mouse-ul pe ele. Limbajul HTML nu oferă instrumentele necesare pentru a specifica ce anume se întâmplă când butonul este apăsat. Pentru aceasta este necesar să includeţi în documentul HTML scripturi JavaScript care să trateze aceste evenimente.

Ca şi butoanele *Submit* şi *Reset*, butoanele obişnuite sunt incluse în formular cu ajutorul etichetei `<INPUT>` având atributul `type` setat la valoarea `button`:

```
<INPUT type="button" name="buton" value="text">.
```

Textul specificat ca valoare a atributului `value` va fi afişat pe buton.

Butoanele pot fi folosite pentru a valida informaţiile introduse în formular, pentru a deschide documente şi a iniţia diverse alte acţiuni din partea browserului.

Câmpuri ascunse (hidden fields)

Câmpurile ascunse sunt elemente care fac parte dintr-un formular dar nu sunt vizibile în cadrul paginii. Prin intermediul acestora se pot include în formular informaţii care nu pot fi alterate de browser sau de utilizator. Introducerea unui astfel de câmp în formular se face prin eticheta
`<INPUT type="hidden">`.
Ca şi în cazul celorlalte elemente ale formularului este necesară prezenţa atributelor `name` şi `value`, ca în exemplul următor:

```
<INPUT type="hidden" name="ascuns" value="modifica">
```

Câmpurile ascunse pot servi la o gestionare mai bună a datelor transmise la server. De exemplu, să presupunem că avem un formular care cere utilizatorului câteva informaţii iniţiale: numele şi adresa. Pe baza acestora, aplicaţia de pe server iniţiază afişarea unui nou formular care solicită introducerea unor informaţii mai detaliate.
Deoarece este plictisitor pentru vizitator să reia introducerea informaţiilor iniţiale, scriptul poate fi configurat să depună primele informaţii în câmpurile ascunse ale celui de-al doilea formular. În această situaţie este utilă folosirea câmpurilor ascunse deoarece serverul procesează un singur formular la un moment dat şi nu are cum să ştie că cel de-al doilea formular a fost completat de aceeaşi persoană.

4. Etichetele `<SELECT>` *şi* `<OPTION>`

Cu ajutorul etichetei **`<SELECT> </SELECT>`** se poate introduce într-un formular un meniu derulant. Crearea unui meniu pentru vizitatorii paginii îi ajută la selectarea unor opţiuni dintr-o listă predefinită.
Fiecare opţiune care face parte din blocul `<SELECT>` se introduce prin eticheta **`<OPTION>`**.

În Exemplul 11.5 este construit un meniu derulant, din care vizitatorul poate selecta opţiunea dorită. Atributul **`name`** are acelaşi rol ca şi în eticheta `<INPUT>`. Dacă vizitatorul selectează din meniu zodia Taur, de exemplu, la apăsarea butonului de expediere, serverului îi este transmisă perechea: `"zodia=taur"`. Atributul **`size`** este setat la valoarea `"1"`. Setarea atributului `size` la valoarea `"1"` creează o un meniu derulant cu o singură opţiune vizibilă iniţial aşa cum se poate observa din Figura 11.5.

Exemplul 11. 5

```
<HTML>
  <HEAD>
    <TITLE>formulare5</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Meniu derulant</H1><HR>
    <FORM action="mailto:adresa@yyy.com" method="post">
      Care este zodia dumneavoastra?<BR>
      <SELECT name="zodia" size="1">
        <OPTION>Berbec
        <OPTION>Taur
        <OPTION>Gemeni
        <OPTION>Rac
        <OPTION>Leu
        <OPTION>Fecioara
        <OPTION>Balanta
        <OPTION>Scorpion
        <OPTION>Sagetator
        <OPTION>Capricorn
        <OPTION>Varsator
        <OPTION>Pesti
      </SELECT>
      <INPUT type="submit" value="expediaza">
    </FORM>
  </BODY>
</HTML>
```

Dacă atributul `size` este setat la o valoare mai mare decât 1 meniul va afişa un număr de opţiuni egal cu această valoare şi va conţine o bară de derulare pentru a face accesibile şi celelalte opţiuni. În Figura 11.6 poate fi văzut un meniu la care valoarea atributului `size` este 6.

5. Eticheta `<TEXTAREA>`

Cu ajutorul etichetei **`<TEXTAREA> </TEXTAREA>`** puteţi insera în pagină o casetă de text multi-linie care permite vizitatorului să introducă un text mai lung, care se poate întinde pe mai multe linii. Această etichetă se foloseşte atunci când dorim să cerem vizitatorului o opinie despre un anumit subiect, care necesită introducerea unui text mai lung decât o singură linie (acest element se mai numeşte şi casetă de comentarii). `<TEXTAREA>` este o etichetă container, deci necesită eticheta de închidere `</TEXTAREA>`.

Atributele **`rows`** şi **`cols`** stabilesc numărul de linii, respectiv de coloane rezervate pentru introducerea textului.
Exemplul 11.6 introduce în pagină o casetă pentru comentarii, aspectul acesteia fiind redat în Figura 11.7.

Exemplul 11. 6

```
<HTML>
  <HEAD>
    <TITLE>formulare6</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Caseta de comentarii</H1><HR>
    <FORM action="mailto:adresa@yyy.com" method="post">
      Care este opinia dumneavoastra despre acest film?<BR>
      <TEXTAREA name="comentariu" rows="10" cols="40"></TEXTAREA><BR>
      <INPUT type="submit" value="expediaza">
    </FORM>
```

```
  </BODY>
</HTML>
```

Toate elementele prezentate pot fi reunite într-un singur formular sau puteţi crea formulare diferite, în funcţie de informaţiile pe care doriţi să le obţineţi de la vizitatorii paginii.

Dacă doriţi să introduceţi în pagină mai multe formulare, sau elemente de tipuri diferite în cadrul aceluiaşi formular, trebuie să aveţi în vedere aspectele care ţin de alinierea acestora. O posibilitate de a alinia elementele conţinute într-un formular este folosirea tabelelor.

Exemplul 11.7 conţine elemente de mai multe tipuri încadrate într-un formular unic. Pentru alinierea unora dintre ele am folosit un tabel. Aspectul paginii descrise în acest document este cel din Figura 11.8.

Exemplul 11. 7

```
<HTML>
  <HEAD>
    <TITLE>formulare7</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Alinierea elementelor</H1><HR>
    <FORM action="mailto:adresa@yyy.com" method="post">
      <H3>Comandati Pizza</H3><P>
      <P>Cate pizza doriti?
      <INPUT name="nrpizza" value="0" size=3 maxlength=3><P>
      <INPUT type="radio" name="marime" value="mare" checked>Mare<BR>
      <INPUT type="radio" name="marime" value="medie">Medie<BR>
      <INPUT type="radio" name="marime" value="normala">Normala
      <P>Topping-uri<P>
      <INPUT type="checkbox" name="topping" value="salam">Salam<BR>
      <INPUT type="checkbox" name="topping" value="carnati">Carnati<BR>
      <INPUT type="checkbox" name="topping" value="sunca">Sunca<BR>
      <INPUT type="checkbox" name="topping" value="ciuperci">Ciuperci<BR>
      <INPUT type="checkbox" name="topping" value="ceapa">Ceapa<BR>
      <INPUT type="checkbox" name="topping" value="masline">Masline
      <P>
      <TABLE width="60%">
        <TR>
          <TD width="20%">Numele
          <TD><INPUT type="text" name="nume">
        <TR>
          <TD width="20%">Telefonul
          <TD><INPUT type="text" name="telefon">
        <TR>
          <TD width="20%">Adresa
          <TD><TEXTAREA name="adresa" rows=6 cols=50></TEXTAREA>
        <TR>
          <TD width="20%">Numărul cartii de credit
          <TD><INPUT type="password" name="creditcard" size=20>
      </TABLE>
      <P>
      <CENTER>
        <INPUT type="submit" value="Trimite comanda">
      </CENTER>
    </FORM>
  </BODY>
</HTML>
```

6. Validarea datelor

Unul dintre aspectele importante pe care trebuie să le aveţi în vedere atunci când folosiţi formulare în paginile dumneavoastră este validarea datelor introduse de vizitatorii lor. Limbajul HTML oferă relativ puţine instrumente pentru îndeplinirea acestei sarcini. Totuşi, aveţi la dispoziţie câteva posibilităţi.

Folosiţi atributul `maxlength` atunci când introduceţi un câmp de editare, pentru a împiedica vizitatorul să introducă un număr eronat de caractere. De exemplu, dacă cereţi introducerea într-o casetă de text a codului numeric personal, acea casetă nu trebuie să permită introducerea a mai mult de 13 caractere, un câmp de editare care cere introducerea codului poştal nu trebuie să permită introducerea a mai mult de 4 caractere, etc.

Pentru a vă asigura că vizitatorul introduce date corecte într-un câmp de editare (de exemplu o adresă de mail validă trebuie să conţină obligatoriu caracterul "@") nu există instrumente HTML. Pentru a realiza astfel de sarcini sunt necesare scripturi JavaScript. În capitolul **JavaScript** veţi găsi scripuri utile care fac posibile validările datelor, încă din momentul introducerii lor.

Folosiţi de câte ori este posibil butoanele radio, casetele de validare şi meniurile, pentru a simplifica procesul de introducere a datelor. Fiţi cât mai explicit, asiguraţi indicaţiile necesare pentru completarea formularelor, precizând, unde este cazul, dacă pot fi selectate mai multe opţiuni sau numai una singură.

Atenţie!

Asiguraţi-vă că scriptul care prelucrează informaţiile din formular are facilităţi de gestionare a erorilor.
Dacă este semnalată o eroare, furnizaţi vizitatorului un mesaj de eroare explicit şi, eventual, câteva corecţii posibile.

Una dintre cele mai frustrante situaţii este ca după completarea unui formular complex, la apariţia unei erori, vizitatorul să fie nevoit să reia completarea integrală a formularului. Evitaţi acest lucru utilizând scripturi care returnează utilizatorului formularul completat, având câmpurile greşite marcate pentru corectare.

Nu uitaţi un amănunt foarte important: informaţiile introduse de vizitatori în formulare vă sunt necesare dumneavoastră, pentru a îmbunătăţi eficienţa şi calitatea site-ului sau chiar pentru a derula afaceri prin intermediul său. De aceea corectitudinea acestor date este esenţială.
Faceţi în aşa fel încât vizitatorii site-ului dumneavoastră să completeze uşor şi cu plăcere formularele prezente în pagini.

7. Rezumat

Formularele servesc la stabilirea unei legături interactive cu vizitatorii unei pagini Web şi la obţinerea de informaţii de la aceştia.

Un formular se introduce în pagină cu ajutorul etichetei `<FORM> </FORM>`.

În cadrul acestei etichete pot fi incluse diverse elemente care fac parte din formular:

- prin eticheta <INPUT> pot fi introduse câmpuri de editare, câmpuri de tip *password*, butoane radio, casete de selecţie, butoane de expediere, butoane de anulare şi butoane obişnuite
- prin eticheta <SELECT> pot fi introduse meniuri derulante, fiecare element al meniului fiind introdus prin eticheta <OPTION>
- prin eticheta <TEXTAREA> pot fi introduse câmpuri de editare multilinie.

Pentru a specifica unde vor fi trimise datele introduse în formular se foloseşte atributul `action` al etichetei <FORM>. Pentru a stabili modul în care aceste date sunt transmise se foloseşte atributul `method` al etichetei <FORM>.

Formularele reprezintă partea vizibilă a procesului de interacţiune cu vizitatorul paginii Web. Odată ce datele au fost introduse în câmpurile formularului, ele trebuie transmise, prelucrate şi stocate, în funcţie de nevoile creatorului paginii. Acest lucru se realizează cu ajutorul scripturilor CGI instalate pe server, despre care vom vorbi în capitolul următor.

8. Test

1. *Ce atribut indică browserului unde să trimită informaţiile din formular?*

a) `input`
b) `action`
c) `method`

2. *Care dintre aceste valori nu aparţine atributului* `method`?

a) `Get`
b) `Href`
c) `Post`

3. *Care dintre următoarele afirmaţii este falsă?*

a) Prin eticheta <INPUT> se pot introduce în formular câmpuri de editare de tip *password*.
b) Prin eticheta <INPUT> se pot introduce în formular butoane radio.
c) Prin eticheta <INPUT> se pot introduce în formular meniuri derulante.

4. *Pentru a introduce în formular un buton de expediere se foloseşte construcţia:*

a) `<INPUT type="reset">`
b) `<INPUT type="submit">`
c) `<INPUT type="forward">`

5. *Fie următoarea construcţie:*

```
<FORM action="mailto:adresa@yyy.com" method="post">
  <INPUT type="radio" name="limba" value="romana">Limba romana
  <INPUT type="radio" name="limba" value="germana">Limba germana
  <INPUT type="radio" name="limba" value="maghiara">Limba maghiara
</FORM>
```

Dacă este bifată opţiunea Limba romana, care este perechea nume/valoare care va fi transmisă serverului?

a) `"limba=Limba romana"`
b) `"limba=romana"`

c) `"radio=romana"`

6. *Care este diferenţa dintre un câmp de editare de tip text şi unul de tip password?*

a) Într-un câmp de tip text se pot introduce mai multe linii, într-unul de tip *password* o singură linie.
b) Într-un câmp de tip text caracterele introduse sunt afişate, într-unul de tip *password* sunt înlocuite cu asteriscuri.
c) Câmpul de editare se introduce cu ajutorul etichetei `<INPUT>`, câmpul de tip *password* se introduce cu ajutorul etichetei `<SELECT>`.

7. *Pentru a insera într-un formular o casetă de validare se foloseşte eticheta* `<INPUT>` *cu atributul* `type` *setat la valoarea:*

a) `"radio"`
b) `"checkbox"`
c) `"option"`

8. *Ce realizează eticheta* `<SELECT>`*?*

a) Introduce un buton de anulare.
b) Introduce un meniu derulant.
c) Introduce o listă de casete de validare.

9. *La ce serveşte eticheta* `<TEXTAREA>`*?*

a) Stabileşte zona din fereastra browserului care este rezervată pentru text.
b) Introduce în formular un câmp de editare multi-linie.
c) Introduce în formular un câmp de editare pe o singură linie.

10. *Care dintre următoarele afirmaţii este falsă?*

a) Datele introduse în formular sunt trimise la server, dacă este folosit un script CGI.
b) Datele introduse în formular sunt trimise la o adresă de mail specificată, dacă este folosită comanda `mailto:`
c) Datele introduse în formular sunt stocate pe hard-disk-ul vizitatorului paginii.

Formulare

1. b)
2. b)
3. c)
4. b)
5. b)
6. b)
7. b)
8. b)
9. b)
10. c)

Capitolul 12
Scripturi CGI

Programare CGI, scripturi CGI sunt cuvinte care stârnesc, cel mai adesea, teama. Multora dintre cei care doresc să creeze pagini Web şi nu sunt programatori profesionişti această zonă le apare misterioasă şi dificil de înţeles. În acest capitol vom învăţa chestiunile fundamentale şi principiile care guvernează realizarea acestor scripturi precum şi modul lor de utilizare. Veţi constata că, deşi realizarea unui script este destul de laborioasă şi necesită cunoaşterea aprofundată a cel puţin unui limbaj de programare (C++ sau Perl), nu vă va fi deloc greu să înţelegeţi modul cum sunt ele concepute.

Vestea bună este aceea că, deşi stăpânirea unui limbaj de programare este binevenită, nu este absolut necesară pentru a putea utiliza scripturi CGI în paginile dumneavoastră. Nu este nevoie să creaţi propriile dumneavoastră scripturi. Pe Web există mii de scripturi gratuite care pot fi descărcate şi utilizate conform propriilor necesităţi. Tot ceea ce aveţi de făcut este să înţelegeţi principiile după care sunt realizate, modul lor de funcţionare şi maniera în care trebuie să le setaţi pentru a le adapta nevoilor dumneavoastră.

1. Ce este CGI?

Să începem prin a preciza ce **nu** este **CGI**: nu este un limbaj de programare. CGI, prescurtare de la *Common Gateway Interface*, este un **protocol standard de comunicare** între documentele Web şi aplicaţiile localizate pe serverul Web.

Scripturile CGI sunt **programe** care respectă acest protocol (un protocol este un set de reguli). Un script CGI este, deci, un program care comunică într-un anumit mod cu pagina dumneavoastră. Existenţa acestui protocol de comunicare între programele de pe server şi documentele Web permite crearea unor pagini interactive şi dinamice, lucru care nu poate fi făcut folosind doar HTML.

Atunci când browserul solicită un script CGI aflat pe server, serverul lansează în execuţie scriptul şi îi transmite acestuia headerele HTTP de cerere primite de la browser. După ce execuţia scriptului se încheie, rezultatele sunt transmise serverului, care formatează headerele de răspuns şi le transmite browserului pentru ca acesta să afişeze rezultatele. O altă posibilitate este ca scriptul să conţină instrucţiuni prin care headerele de răspuns sunt configurate chiar de script şi transmise de acesta direct browserului.

Indiferent dacă solicită un document Web sau un script, browserul trebuie să cunoască locaţia serverului Web şi numele fişierului solicitat. Această informaţie îi este transmisă browserului prin intermediul atributului `action` al etichetei `<FORM>` care primeşte drept valoare adresa URL a scriptului stocat pe server. De obicei, scripturile CGI sunt stocate pe server într-un director special destinat lor, care se numeşte ***cgi-bin***.

2. Transferul datelor

Aşa cum am văzut în capitolul anterior, datele pe care dorim să le obţinem de la vizitatorul paginii sunt colectate prin intermediul formularelor. Datele astfel colectate sunt transmise de către browser la serverul pe care este instalat scriptul CGI prin intermediul headerelor HTTP de cerere.

Cele mai importante headere HTTP de cerere sunt **`Get`** şi **`Post`**. Prin intermediul atributului `method` al etichetei `<FORM>` este specificat ce tip de header HTTP de cerere este utilizat pentru a transfera datele la server. Dacă metoda folosită este `Get`, datele sunt transmise prin intermediul adresei URL. Dacă folosiţi metoda `Post` datele sunt transmise sub forma unui fişier separat.

Este recomandată folosirea metodei `Post` atunci când volumul de date transmise este mare (depăşeşte 1024 de octeţi, lungimea maximă a unui URL). După ce metoda de transmitere este stabilită, browserul creează un header HTTP de cerere prin care este identificată localizarea scriptului pe server şi apoi transmite headerul la server. Serverul recepţionează headerul de cerere şi apelează scriptul CGI.

Odată datele ajunse la server, scriptul este lansat în execuţie iar datele îi sunt transmise acestuia prin intermediul unui tip special de variabile numite *variabile de mediu* (dacă folosiţi metoda `Get`) sau prin intermediul fişierului standard de intrare (dacă folosiţi metoda `Post`).

Scriptul CGI execută sarcinile pentru care a fost conceput şi obţine anumite rezultate care urmează a fi transmise browserului prin intermediul unor headere HTTP de răspuns. De obicei, scriptul configurează un singur header de răspuns, şi anume `Content-Type`. Acest header comunică browserului tipul de date care îi vor fi returnate: documente HTML, imagini sau alt tip de fişiere. Serverul adaugă şi el headere HTTP de răspuns suplimentare, apoi strânge toate rezultatele şi headerele de răspuns într-un pachet care este transmis browserului. Browserul îl recepţionează şi afişează informaţiile primite în modul descris de headerele de răspuns.

3. Execuţia scriptului

Ce face scriptul CGI după ce este lansat în execuţie? Răspunsul este simplu: execută sarcinile pentru care a fost construit. Sarcinile pe care le poate realiza un script sunt diverse şi numeroase. Iată câteva exemple:

- manipularea informaţiilor introduse prin intermediul formularelor
- manipularea hărţilor de imagini
- generarea contoarelor care monitorizează numărul de accesări ale paginii (*hit counters*)
- construirea de motoare de căutare
- administrarea licitaţiilor *on-line*
- crearea de aplicaţii interactive cum ar fi forumurile sau camerele de chat
- crearea şi manipularea bazelor de date

Cum am mai spus, scrierea unui script care îndeplineşte astfel de sarcini nu este simplă. Pentru a putea concepe un script este necesară cunoaşterea aprofundată a unui limbaj de programare. Sunt utilizate limbaje variate pentru realizarea scripturilor: Perl, C sau C++, Visual Basic, etc. Descrierea acestor limbaje şi a principiilor de programare depăşeşte cu mult obiectul acestei cărţi. De altfel, aşa cum spuneam, pentru a construi o pagina Web interactivă cu ajutorul scripturilor CGI, nu este necesar să le creaţi chiar dumneavoastră. Pe Web există foarte multe site-uri care pun la dispoziţia vizitatorilor mii de scripturi gratuite. De obicei ele sunt însoţite de instrucţiuni de instalare şi utilizare. Puteţi alege dintre acestea pe acelea care satisfac cerinţele dumneavoastră. Totuşi, pentru a face o alegere în cunoştinţă de cauză precum şi pentru a putea să faceţi modificările necesare instalării şi funcţionării scriptului, aveţi nevoie să cunoaşteţi câteva noţiuni legate de structura unui script CGI.

4. Structura unui script

Structura unui script CGI conţine următoarele secţiuni:

- citirea datelor introduse prin intermediul formularului
- prelucrarea datelor şi efectuarea sarcinilor impuse de programator
- afişarea rezultatelor.

Citirea datelor

Unul dintre elementele de bază cu care lucrează un script îl constituie **variabilele de mediu**. Variabilele de mediu sunt o categorie specială de variabile, folosite de server în procesul de execuţie al scriptului. Aceste variabile conţin informaţii despre server, despre browser şi despre datele introduse de utilizator. Numărul variabilelor de mediu este destul de mare, dar există câteva care au o importanţă specială pentru script.

Acestea sunt:

REQUEST_METHOD
QUERY_STRING
CONTENT_LENGTH

Variabila de mediu **REQUEST_METHOD** poate avea cele două valori despre care am mai amintit: `Get` şi `Post`. Prin intermediul ei serverul informează scriptul asupra modului cum îi sunt transmise datele.
Aşa cum am învăţat în capitolul anterior, datele introduse de utilizator sunt transmise în perechi nume/valoare.
Variabila de mediu **QUERY_STRING** este folosită pentru a reţine datele de intrare în cazul folosirii metodei `Get` iar variabila **CONTENT_LENGTH** este folosită pentru a specifica lungimea şirului de caractere citit din fişierul de intrare, în cazul folosirii metodei `Post`.

Transmiterea datelor prin metoda `Get`

Când serverul foloseşte metoda **`Get`** datele introduse în formular sunt adăugate la sfârşitul adresei URL a scriptului.

Pentru a face lucrurile mai clare, să luam un exemplu. să presupunem că în pagina dumneavoastră se află un formular care conţine două câmpuri de editare numite `"nume"` şi `"prenume"`. Cu alte cuvinte codul formularului dumneavoastră va arăta astfel:

```
<FORM action="cgi-bin/script.cgi" method="Get">
  <INPUT type="text" name="nume">Introduceti numele
  <INPUT type="text" name="prenume">Introduceti prenumele
  <INPUT type="submit">
</FORM>
```

Să mai presupunem că vizitatorul introduce în câmpul `"nume"` valoarea `"Popescu"` iar în câmpul `"prenume"` valoarea `"Ion"` şi că adresa URL a scriptului nostru este www.html.com/cgi-bin/scriptulmeu.cgi/

Atunci serverului îi este transmisă următoarea adresă URL:

```
http://www.html.com/cgi-bin/scriptulmeu.cgi/?nume=Popescu&prenume=Ion
```

Şirul de caractere de după semnul de întrebare este depus în variabila de mediu `QUERY_STRING`.

Transmiterea datelor prin metoda `Post`

Metoda **`Post`** este cea mai folosită în transmiterea datelor. Atunci când datele sunt trimise prin această metodă, ele sunt furnizate scriptului prin intermediul **fişierului standard de intrare**, **STDIN**. Pentru a preciza scriptului lungimea şirului de caractere (numărul de octeţi) pe care îl are de citit, aceasta este reţinută în variabila de mediu `CONTENT_LENGTH`.

Indiferent de metoda folosită, după citirea datelor scriptul trebuie să le decodifice, adică să elimine din şirul de caractere primit toate caracterele care nu au fost introduse de utilizator: "&", "=" şi altele.

După citirea şi decodificarea datelor scriptul execută instrucţiunile de procesare a lor şi pregăteşte răspunsul pentru server.

Afişarea rezultatelor

În general, răspunsurile pe care scriptul le transmite serverului se împart în două categorii:

- Sarcina a fost îndeplinită cu succes
- Sarcina nu a fost îndeplinită, au apărut erori.

Dacă execuţia scriptului este încununată de succes iar sarcinile sale au fost îndeplinite, acesta transmite serverului un răspuns în consecinţă. Dacă, din diverse motive, au apărut erori la execuţia scriptului iar acesta nu şi-a dus sarcinile la bun sfârşit, serverului îi este prezentat un mesaj de eroare, care trebuie să conţină informaţii despre natura erorii apărute.

De exemplu, să presupunem că în pagina dumneavoastră aţi plasat un formular care solicită datele personale ale vizitatorului, care sunt transmise unui script a cărui sarcină este să trimită aceste date la o adresă specificată de mail (aceasta este una dintre cele mai comune utilizări ale scripturilor). După ce vizitatorul a introdus datele iar scriptul şi-a încheiat execuţia, este bine să afişeze un mesaj pentru vizitator prin care îl informează că datele sale au fost expediate. Pe de altă parte, dacă a survenit o eroare la introducerea datelor, vizitatorului trebuie să-i fie oferit un mesaj prin care este informat despre acest lucru şi, în plus, despre posibilitatea de a remedia eroarea.

Mesajele privind rezultatele execuţiei pot fi trimise de script serverului, care formatează headerele HTTP de răspuns, sau scriptul poate formata el însuşi headerele şi poate trimite rezultatele direct la browser.

Există trei tipuri fundamentale de headere de răspuns:

1. **`Content-Type`**
2. **`Location`**
3. **`Status`**

`Content-type`

Browserul dumneavoastră primeşte headere de tipul `Content-Type` ori de câte ori primeşte de la server un document HTML, o imagine sau orice alt tip de fişier. Indiferent de natura fişierului transmis de la server la browser, serverul va

asocia fişierului acest header pentru a comunica browserului tipul fişierului trimis.

Tipurile de fişiere care pot fi transmise browserului sunt asociate sub numele de tipuri **MIME** (*Multipurpose Internet Mail Extensions*). Clasificarea se bazează pe conţinutul fişierului trimis. Iată cele şapte tipuri MIME fundamentale:

- Text
- Multipart
- Message
- Application
- Image
- Audio
- Video

Fiecare dintre aceste tipuri are un număr de sub-tipuri. Headerul `Content-Type` specifică atât tipul cât şi subtipul MIME al fişierului trimis. Iată câteva exemple:

- Text/HTML
- Text/plain
- Image/gif
- Image/jpeg

`Location`

Acest header conţine adresa URL a documentului pe care scriptul doreşte să-l trimită ca răspuns către browser. De exemplu, dacă răspunsul pe care doriţi să-l primească vizitatorul paginii după transmiterea datelor este un document HTML care se numeşte *multumesc.html* atunci scriptul va conţine un header ca acesta:

```
Location: multumesc.html
```

urmat de o linie liberă. În acest mod browserul este redirecţionat către URL-ul documentului respectiv. În situaţia când folosiţi headerul `Location` pentru a afişa răspunsul, scriptul nu va mai include headerul de răspuns `Content-Type`.

`Status`

Headerul `Status` returnează către browser un cod format din trei cifre împreună cu un text explicativ. Aţi văzut de foarte multe ori acest header în situaţiile când cererea trimisă de browser nu a fost îndeplinită, iar documentul solicitat nu a putut fi furnizat de către server. Iată câteva dintre cele mai cunoscute coduri de status:

- 400 – Bad request – apare în situaţia când sintaxa headerului HTTP de cerere a fost eronată
- 401 – Unauthorized – apare când, pentru a obţine documentul solicitat, sunt necesare anumite privilegii de acces
- 403 – Forbidden – apare când serverul refuză accesul la document
- 404 – File Not Found – apare în situaţia când serverul nu poate găsi documentul solicitat

5. Exemple de scripturi

Limbajul cel mai folosit pentru scrierea scripturilor este Perl. Chiar dacă nu cunoaşteţi acest limbaj, dacă aveţi totuşi cunoştinţe minime de programare, nu vă va fi greu să înţelegeţi structura scripturilor prezentate în continuare.

Din păcate, spre deosebire de documentele HTML care pot fi scrise într-un simplu editor de text şi apoi vizualizate cu ajutorul browserului, cu scripturile lucrurile nu mai stau aşa.

Perl este un limbaj interpretat. Aceasta înseamnă că puteţi scrie codul sursă într-un editor de text, cum ar fi Notepad. Dar pentru ca scriptul să funcţioneze, serverul trebuie să aibă instalat un interpretor de Perl. Majoritatea serverelor Web deţin un astfel de interpretor. Dacă doriţi să instalaţi şi dumneavoastră un interpretor Perl şi să testaţi funcţionarea scripturilor prezentate în exemplele de mai jos, ba mai mult, dacă doriţi să vă transformaţi propriul calculator într-un server Web pe care să puteţi rula scripturi Perl, precum şi să vă testaţi funcţionarea întregului site, veţi afla cum puteţi realiza acest lucru în ultima parte a acestui capitol.

Pentru moment, iată câteva exemple simple de scripturi Perl din care vă puteţi face o idee despre structura unui script şi sintaxa limbajului Perl. Cel mai simplu script este clasicul exemplu "*Hello, world !*"prezentat în Exemplul 12.1. Acest script ilustrează modul în care se realizează configurarea headerului de răspuns `Content-Type` şi afişarea unui document HTML.

Exemplul 12. 1

```
1          #!/usr/bin/perl
2
3          print "Content-Type: text/html\n\n";
4
5          print "<HTML>\n";
6          print "<HEAD>\n";
7          print "<TITLE>Hello World !</TITLE>\n";
8          print "</HEAD>\n";
9          print "<BODY>\n";
10         print "<H1>Hello World !</H1>\n";
11         print "<P>\n";
12         print "<H3>Have a nice day</H3>\n";
13         print "</BODY>\n";
14         print "</HTML>\n";
```

Liniile scriptului nu sunt, în general, numerotate, numerotarea servind aici doar la referirea lor mai uşoară. Acest exemplu simplu realizează afişarea pe ecran a textului "Hello World !", şi a mesajului "Have a nice day". Nu există o etapă de citire a datelor, deoarece nu există date de intrare.

Linia 1 conţine calea spre interpretorul Perl aflat pe server. Dacă sistemul sub care lucrează serverul este UNIX, semnul # trebuie îndepărtat. În linia a 3-a este configurat headerul HTTP de răspuns, `Content-Type`. Aşa cum observaţi, documentul care va fi transmis browserului este de tip text, şi anume un document HTML.

Secvenţa de caractere `"\n"` care se repetă la finalul fiecărei linii se numeşte **secvenţă escape**. Rolul acesteia este să realizeze, la afişare, trecerea pe rândul următor. Pe linia în care este definit headerul `Content-Type` este obligatorie prezenţa a două secvenţe escape.

Toate instrucţiunile scriptului sunt instrucţiuni de afişare. Prin intermediul lor, scriptul descrie browserului pagina Web pe care urmează să o afişeze. Structura paginii conţine, după cum vedeţi, toate secţiunile necesare: eticheta de deschidere a documentului: `<HTML>`, antetul: `<HEAD>` şi corpul documentului: `<BODY>`.

Dacă doriţi să testaţi funcţionarea scriptului, scrieţi-l în Notepad şi salvaţi-l sub numele *hello.pl* (extensia **.pl** indică un program scris în limbajul Perl. Se mai poate folosi şi extensia **.bgi**). Apoi instalaţi scriptul în directorul *cgi-bin* al serverului dumneavoastră.
După executarea scriptului, pagina de răspuns arată ca în Figura 12.1

Exemplul următor realizează preluarea datelor introduse de vizitator prin intermediul unui formular şi afişarea lor pe ecran. Formularul pentru introducerea datelor prezentat în Exemplul 12.2 conţine doar casete de text, în care vor fi introduse numele, prenumele şi adresa de mail. Atributul `action` al etichetei `<FORM>` trimite la scriptul *script2.pl* aflat în directorul *cgi-bin* al serverului. Aspectul formularului este cel din Figura 12.2

Exemplul 12. 2

```
<HTML>
  <HEAD>
    <TITLE>cgi2</TITLE>
  </HEAD>
  <BODY>
    <FORM method="POST" action="cgi-bin/script2.pl">
      Numele:
      <INPUT type="text" name="nume"><BR>
      Prenumele:
      <INPUT type="text" name="prenume"><BR>
      Adresa de email:
      <INPUT type="text" name="email"><BR>
      <INPUT type="submit" value="Trimite">
    </FORM>
  </BODY>
</HTML>
```

Observaţi că datele sunt transmise la scriptul *script2.pl* aflat în directorul *cgi-bin*, prin intermediul metodei `Post`. Exemplul 12.3 ilustrează modul de construire al scriptului *script2.pl*.

Exemplul 12. 3

```
1          #!/usr/bin/perl
2
3          # Plasarea datelor transmise prin metoda Post in variabile
4
5          read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
6          @pairs=split(/&/, $buffer);
7          foreach $pair (@pairs)
8          {
9          ($name, $value)=split(/=/, $pair);
10         $value=~ tr/+/ /;
11         $value=~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;

13         $FORM{$name}=$value;
14         }
15         #afisarea rezultatului
16
17         print "Content-type: text/html\n\n";
18
19 print "<HTML>\n";
20 print "<HEAD>\n";
21 print "<TITLE>Multumesc!</TITLE>\n";
22 print "</HEAD>\n";
```

```
23 print "<BODY>\n";
24 print "<H1>Multumesc!</H1>\n";
25 print "<BR>\n";
26 print "<H4>Datele introduse sunt:</h4>\n";
27 print "<BR>\n";
28 print "<H3>$FORM{'nume'} $FORM{'prenume'}</H3>\n";
29 print "<H3>$FORM{'email'}</H3>\n";
30 print "</BODY>\n";
31 print "</HTML>\n";
```

Se impun câteva explicaţii. Mai întâi, observaţi pe liniile 1, 3 şi 15 prezenţa caracterului #. Acesta este modul de a introduce comentarii într-un program Perl. Comentariile sunt extrem de utile, mai ales la scripturi. Un script necomentat este foarte greu de utilizat. Ca şi în cazul comentariilor introduse în documentele HTML, acestea sunt ignorate la execuţie.

Prima secţiune a scriptului realizează citirea datelor. Deoarece acestea au fost transmise prin metoda `Post`, observaţi că citirea se face din fişierul standard de citire, STDIN. Între liniile 5 şi 14 se realizează citirea, decodificarea datelor şi plasarea lor în variabilele corespunzătoare. Secvenţa dintre liniile 16 şi 31 reprezintă partea de afişare a rezultatelor. Ca şi în exemplul anterior, observaţi că linia în care se face formatarea headerului de răspuns `Content-Type` necesită prezenţa a două secvenţe escape `"\n\n"`.

Să presupunem că datele introduse în formular sunt:

- nume - Ionescu
- prenume - Maria
- adresa de mail - ionmar@yahoo.com.

Atunci pagina generată de script va arata ca în Figura 12.3

Apelul scripturilor nu se face numai prin intermediul formularelor. Se poate realiza şi prin intermediul imaginilor sau al legăturilor. Prezentăm în continuare o pagina Web care apelează scriptul *ip.pl* prin intermediul unei legături. Scriptul realizează afişarea adresei IP a computerului pe care este instalat.

Exemplul 12.4 prezintă documentul HTML în care este apelat scriptul. Aspectul paginii este cel din Figura 12.4

Exemplul 12. 4

```
<HTML>
  <HEAD>
    <TITLE>cgi4</TITLE>
  </HEAD>
  <BODY>
    <FONT face="arial" size="4">
      Pentru a afla adresa dumneavoastra de IP apasati
      <A href="cgi-bin/ip.pl"> aici</A>
    </FONT>
  </BODY>
</HTML>
```

Scriptul *ip.pl* foloseşte variabila de mediu `REMOTE_ADDR` pentru a obţine adresa IP. În Exemplul 12.5 este prezentat scriptul *ip.pl*.

Exemplul 12. 5

```
1          #!/usr/bin/perl
2
```

```
3          print "Content-Type: text/html\n\n";
4
5          print "<HTML>\n";
6          print "<HEAD>\n";
7          print "<TITLE>cgi3</TITLE>\n";
8          print "</HEAD>\n";
9          print "<BODY>\n";
10         print "<P>\n";
11         print "<H3>Adresa IP a computerului dvs. este:
$ENV{REMOTE_ADDR} </H3>\n";
12         print "</BODY>\n";
13         print "</HTML>\n";
```

La apăsarea link-ului din documentul HTML este apelat scriptul, care afişează pagina de răspuns din Figura 12.5

Evident, dacă veţi rula scriptul pe alt computer, adresa IP obţinută va fi diferită. Când scriptul este apelat printr-un link nu există posibilitatea de a i se transmite date.

6. Instalarea interpretorului Perl şi a aplicaţiei PWS

Pentru a putea verifica funcţionarea exemplelor prezentate precum şi a scripturilor pe care doriţi să le folosiţi în cadrul site-ului dumneavoastră înainte de etapa publicării pe Web trebuie să aveţi instalat un interpretor Perl. Puteţi descărca şi instala interpretorul Active Perl vizitând adresa:

http://www.gossland.com/course/install.html

La aceeaşi adresă puteţi găsi şi aplicaţia Microsoft Personal Web Server (PWS) care este furnizată de Microsoft în pachetele Windows 98 şi Windows 2000. Recomandăm descărcarea aplicaţiei de la adresa menţionată, care are avantajul că include o corecţie a unui *bug* de instalare existent în versiunea oficială Microsoft.

Atenţie!

Microsoft Personal Web Server se poate instala pe computerul personal având rolul de aplicaţie server. Odată instalată, computerul dumneavoastră va funcţiona întocmai ca un server Web, astfel încât vă veţi putea testa funcţionarea scripturilor instalate precum şi funcţionarea întregului site.

Din păcate, Microsoft nu furnizează o aplicaţie server similară şi în pachetele Windows Milenium şi Windows XP iar aplicaţia PWS nu funcţionează decât pe sistemele Windows 98 şi 2000.

La adresa indicată veţi găsi îndrumări precise şi complete asupra tuturor paşilor pe care îi aveţi de urmat pentru a instala atât interpretorul Perl cât şi aplicaţia Personal Web Server. În plus, site-ul respectiv conţine şi un foarte bun curs de iniţiere în limbajul Perl în limba engleză.

7. Rezumat

CGI este un protocol de comunicare face legătura dintre scripturile aflate pe serverul Web şi documentele HTML. Un script CGI este o aplicaţie care respectă acest protocol.

Printre cele mai folosite scripturi CGI sunt cele care servesc la procesarea informaţiilor introduse într-un formular.

Transferul datelor de la formular la serverul pe care se află scriptul se face printr-una dintre cele două metode: `Get` sau `Post`.

După ce datele au fost transmise la server, scriptul le citeşte şi le decodifică, le prelucrează conform instrucţiunilor sale şi transmite un răspuns către browser.

Browserul afişează rezultatele trimise de script în conformitate cu headerele de răspuns incluse în pachetul de date transmis.

Capitolul 13
JavaScript

1. Ce este JavaScript?

JavaScript este un **limbaj de programare** orientat pe obiecte, care vă permite să oferiţi paginilor dumneavoastră un caracter mai dinamic şi interactiv. Dacă aţi parcurs prima parte a cărţii şi aveţi cunoştinţe minime de programare, nu veţi avea dificultăţi în înţelegerea modului în care funcţionează acest limbaj.

Atenţie!

JavaScript nu este acelaşi lucru cu limbajul Java. De fapt, deşi numele sunt asemănătoare, sunt două limbaje foarte diferite. Spre deosebire de Java, care este un limbaj orientat pe obiecte, complex, asemănător cu limbajul C++, JavaScript este mult mai simplu şi mai uşor de folosit. În timp ce Java este un limbaj pentru programatori, JavaScript este destinat neprofesioniştilor care doresc să îmbunătăţească aspectul şi funcţionalitatea paginilor de Web.

JavaScript a fost dezvoltat de Netscape Corporation pentru a fi utilizat în browserul Netscape Navigator. Un script JavaScript poate fi plasat direct în pagina Web, fiind executat de browser împreună cu documentul Web care-l conţine. Browserul Internet Explorer (versiunile mai noi decât IE 3.0) execută şi el corect scripturile JavaScript.

JavaScript este un limbaj interpretat, ca şi Perl, dar în cazul său nu trebuie să vă mai faceţi griji în privinţa instalării unui interpretor deoarece acesta este inclus în browser. Codul sursă poate fi inclus în pagina Web sau poate fi pus într-un fişier separat care este referit din pagină.

Ca şi în cazul scripturilor CGI, nici în acest caz nu este absolut necesară cunoaşterea limbajului JavaScript pentru a putea insera şi utiliza cu succes astfel de scripturi. Există un număr impresionant de scripturi gata de utilizare, care se pot integra cu uşurinţă în codul sursă al documentelor dumneavoastră.

2. Cum poate fi inclus un script în pagină?

Scripturile JavaScript pot fi incluse în pagina Web în două moduri:

- prin intermediul etichetei **`<SCRIPT> </SCRIPT>`**

- prin intermediul unei **proceduri eveniment**

Iată un exemplu foarte simplu în care în document este inclus un script care afişează în pagină textul "Buna ziua!" prin intermediul etichetei <SCRIPT>. Aspectul acestei pagini este cel din Figura 13.1.

Exemplul 13. 1

```
<HTML>
  <HEAD>
    <TITLE>javascript1</TITLE>
  </HEAD>
  <BODY>
    <H1>Primul exemplu JavaScript</H1><HR>
    <SCRIPT language="JavaScript">
      document.write("Buna ziua!")
    </SCRIPT>
  </BODY>
</HTML>
```

Unul dintre atributele etichetei <SCRIPT> este **language** prin care este specificat limbajul în care este scris scriptul, în cazul acesta valoarea atributului fiind "JavaScript".
O altă modalitate de a include de a include un script în pagină este salvarea acestuia ca un fişier extern şi apelarea sa prin intermediul atributului **src** al etichetei <SCRIPT>.

În exemplele de scripturi care vor urma veţi observa diverse moduri în care pot fi incluse în paginile Web scripturi JavaScript.

Nu vom intra în amănunte privind sintaxa limbajului, deoarece aceasta depăşeşte subiectul acestei cărţi. Totuşi, pentru a putea folosi judicios un script JavaScript trebuie să aveţi o imagine despre elementele esenţiale cu care lucrează JavaScript, obiectele şi evenimentele.

Obiecte

Un **obiect** este un nou tip de date, care reuneşte sub aceeaşi denumire atât datele cât şi funcţiile care le prelucrează. Datele se numesc **proprietăţile** obiectului iar funcţiile se numesc **metodele** obiectului.

Pentru a exemplifica aceste concepte, să presupunem că dorim să construim un obiect care să reprezinte o carte de vizită, pe care îl vom numi Card. Acest obiect va avea următoarele proprietăţi:
- nume
- adresă
- telefon

Pentru a defini un obiect se foloseşte o funcţie numită **constructor**. În exemplul nostru, funcţia constructor a obiectului Card va arata astfel:

```
function Card(num, adr, tel)
{
this.nume=num;
this.adresa=adr;
this.telefon=tel;
}
```

În acest moment, constructorul obiectului Card încă nu este complet, el conţine numai proprietăţile obiectului.

Acum să construim şi o metodă care să lucreze cu acest obiect. Deoarece dorim să tipărim cărţile de vizită pe care le vom realiza cu ajutorul acestui obiect, vom crea o funcţie care va tipări proprietăţile obiectului, numită `PrintCard()`.

```
function PrintCard()
{
document.write("Numele: ", this.nume, "\n");
document.write("Adresa: ", this.adresa, "\n");
document.write("Telefonul: ", this.telefon, "\n");
}
```

Acum putem scrie definiţia completă a obiectului nostru:

```
function Card(num, adr, tel)
{
this.nume=num;
this.adresa=adr;
this.telefon=tel;
this.PrintCard=PrintCard;
}
```

Să trecem, acum, la folosirea unui obiect. Pentru a crea un obiect concret, adică o instanţă a obiectului Card, se foloseşte cuvântul cheie `new`. Următoarea instrucţiune construieşte o instanţă a obiectului `Card`, care se numeşte `ana`:

```
ana=new Card ("Ana Munteanu", "Str. Apusului, nr.2, Bucuresti", "6162345")
```

Odată ce a fost creată o nouă instanţă a obiectului `Card`, metoda `PrintCard` poate fi folosită astfel:

```
ana.PrintCard()
```

Şi acum, să punem toate aceste secvenţe de cod laolaltă şi să folosim obiectul într-o pagină Web, pentru a afişa mai multe cărţi de vizită. Exemplul 13.2 realizează acest lucru. Aspectul paginii descrise în document este cel din Figura 13.2.

Exemplul 13. 2

```
<HTML>
  <HEAD>
    <TITLE>javascript2</TITLE>
    <SCRIPT language="JavaScript">
      function PrintCard()
      {
      document.write("<B>Numele:</B> ", this.nume, "<BR>");
      document.write("<B>Adresa:</B> ", this.adresa, "<BR>");
      document.write("<B>Telefonul:</B> ", this.telefon, "<HR>");
      }
      function Card(num, adr, tel)
      {
      this.nume=num;
      this.adresa=adr;
      this.telefon=tel;
      this.PrintCard=PrintCard;
      }
    </SCRIPT>
  </HEAD>
  <BODY>
    <H1 align="center">Crearea obiectelor</H1>
    Inceputul scriptului<HR>
```

```
    <SCRIPT language="JavaScript">
      // Crearea a trei obiecte noi
      ana=new Card ("Ana Munteanu", "Str. Apusului, nr.2, Bucuresti",
      "6162345")
      ion=new Card("Ion Popescu", "Str. Zorilor nr.1, Ploiesti",
      "456789");
      radu=new Card("Radu Ionescu", "Str. Noptii, nr.3, Pitesti",
      "201202");
      // Afisarea lor
      ana.PrintCard();
      ion.PrintCard();
      radu.PrintCard();
    </SCRIPT>
    Sfarsitul scriptului
  </BODY>
</HTML>
```

Puteţi observa modul în care a fost inclus scriptul în pagina Web. Astfel, definiţia scriptului este plasată în antetul documentului HTML, iar apelul scriptului este făcut în corpul documentului. În cadrul scriptului apar două linii precedate de şirul de caractere "//". Acesta este modul în care se introduc comentariile în cadrul scripturilor JavaScript.

Evenimente

În continuare vom discuta despre un alt concept fundamental cu care lucrează JavaScript, şi anume **evenimentul**. Un eveniment este o acţiune care survine la un moment dat şi în urma căreia este declanşată execuţia unei anumite părţi din program. Ori de câte ori vizitatorul face click pe o legătură, de exemplu, când introduce un text sau chiar când trece cu mouse-ul deasupra unei zone a paginii, survine un eveniment la care scriptul reacţionează generând un răspuns.

Iată care sunt tipurile de evenimente cu care lucrează JavaScript:

Tabel 13. 1

Eveniment	Descriere
`onAbort`	Apare când utilizatorul renunţă la încărcarea unei imagini
`onBlur`	Apare când un obiect din pagină pierde focusul
`onChange`	Apare când un câmp de editare este modificat de utilizator (când se introduce un text)
`onClick`	Apare când utilizatorul face click pe un element
`onError`	Apare când un document sau o imagine nu se încarcă corect
`onFocus`	Apare când un element primeşte focusul
`onLoad`	Apare când o pagină sau o imagine îşi termină încărcarea
`onMouseOver`	Apare când cursorul mouse-ului se plimbă deasupra unui element
`onMouseOut`	Apare când cursorul mouse-ului părăseşte elementul

onSelect	Apare când utilizatorul selectează un text
onSubmit	Apare când este apăsat un buton de tip Submit
onUnload	Apare când utilizatorul părăseşte documentul sau sesiunea curentă.

După cum vedeţi, prin intermediul JavaScript se poate răspunde unui mare număr de evenimente. Acest lucru se realizează prin crearea unei **proceduri eveniment**.

Procedurile eveniment nu sunt definite cu ajutorul etichetei <SCRIPT> ci ele sunt atribute ale celorlalte etichete. De exemplu, iată o legătură care include o procedură eveniment:

```
<A href="http://www.amazon.com"
onMouseOver=" window.status='Cea mai mare librarie virtuala'; return
true"> Click aici</A>
```

Observaţi că întreaga procedură eveniment care răspunde la evenimentul MouseOver este inclusă în eticheta <A>. Sunt specificate două instrucţiuni care se execută în momentul când cursorul mouse-ului este plasat deasupra legăturii. Prima afişează un mesaj în bara de status a ferestrei, următoarea returnează valoarea true, pentru a împiedica ştergerea mesajului. Mai observaţi că mesajul care va fi afişat este delimitat de apostrofuri.

Dacă procedura eveniment este mai extinsă, includerea ei în întregime într-o eticheta devine nepractică. În această situaţie, este mai avantajos să construim o funcţie care să trateze evenimentul.
Funcţia este definită în secţiunea <HEAD> a documentului, iar în corpul documentului este apelată ca procedură eveniment. De exemplu, să presupunem că am construit o funcţie eveniment numită Citeste(). Atunci ea poate fi apelată astfel:

```
<A href="#cuprins" onMouseOver="Citeste();">Mergi la Cuprins</A>
```

În momentul când mouse-ul se află deasupra legăturii, funcţia este lansată în execuţie.

3. Modul de execuţie al scriptului

Scripturile integrate în cadrul paginilor sunt evaluate după ce încărcarea paginii s-a încheiat dar înainte ca aceasta să fie afişată. Pe de altă parte, scripturile stocate ca fişiere separate sunt evaluate înaintea tuturor scripturilor *in-line* (adică a celor incluse în pagină). Funcţiile definite în cadrul scripturilor nu sunt executate automat la încărcarea paginilor ci abia atunci când acestea sunt apelate, fie prin eticheta <SCRIPT> fie printr-o procedură eveniment.

Dacă aţi văzut vreodată un script JavaScript într-o pagină Web, poate aţi remarcat un lucru aparent ciudat: întregul script este inclus între etichetele HTML de comentariu:

```
<!- - comentariu - ->.
```

Motivul este faptul că există browsere care nu recunosc şi nu pot executa scripturile JavaScript. Dacă un document Web care conţine cod JavaScript este încărcat într-un asemenea browser, în loc ca scriptul să fie executat, este afişat

în pagină întregul cod, ceea ce nu este de dorit. Aşa după cum ştiţi, comentariile sunt ignorate de browser, astfel că includerea codului JavaScript între etichetele de comentariu va duce la ignorarea acestei părţi a documentului. Excepţie fac browserele care recunosc JavaScript şi care vor identifica prezenţa JavaScript şi vor executa scriptul.

În concluzie, maniera recomandată pentru introducerea în pagină a unui script este următoarea:

```
<SCRIPT language="JavaScript">
<! - -
Cod JavaScript
// - ->
</SCRIPT>
```

Am învăţat până acum cum poate fi inclus codul JavaScript într-un document HTML. Să vedem, mai departe, cum putem folosi scripturile JavaScript pentru a face paginile mai atractive şi a le îmbunătăţi funcţionalitatea.

4. Ce se poate face cu JavaScript?

În continuare sunt prezentate câteva exemple de scripturi cu ajutorul cărora se pot realiza lucruri care depăşesc posibilităţile limbajului HTML, cum ar fi afişarea datei curente în pagina dumneavoastră, deschiderea unei ferestre pop-up, afişarea unui mesaj în bara de status a ferestrei browserului, afişarea aleatoare a unor mesaje în pagină, imagini care îşi schimbă aspectul la trecerea mouse-ului, şi, una dintre cele mai utile aplicaţii JavaScript, validarea formularelor.

Afişarea unui mesaj în bara de status

Acest exemplu ilustrează modul cum se poate afişa un mesaj în bara de status (aflată la baza ferestrei browserului) la trecerea cu mouse-ul peste un anumit element din pagină. De obicei, acest element este un link sau o imagine. În Exemplul 13.3, elementul este o imagine. Aspectul paginii care conţine scriptul este cel din Figura 13.3.

Exemplul 13. 3

```
<HTML>
  <HEAD>
    <TITLE>javascript3</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Mesaj in status bar</H1><HR>
    <H3>Mesajul apare la plasarea mouse-ului pe imagine</H3>
    <IMG src="../Imagini/tiger.gif" onMouseOver="window.status='Acesta
    este un tigru';
    return true" onMouseOut="window.status=' ';return true">
  </BODY>
</HTML>
```

Afişarea datei curente în pagină

Scriptul următor preia data sistemului de operare al utilizatorului şi o afişează în pagină. Evident, dacă data sistemului este setată incorect, ea va apărea în pagină ca atare. Exemplul 13.4 construieşte o pagină Web care conţine un astfel de script, aspectul paginii fiind redat în Figura 13.4.

Exemplul 13. 4

```
<HTML>
  <HEAD>
    <TITLE>javascript4</TITLE>
    <SCRIPT language="javascript">
      <!--
      var zi, data, luna
      azi=new Date()
      if(azi.getDay(==0){ zi="Duminica, " }
      else if(azi.getDay()==1){ zi="Luni, " }
      else if(azi.getDay()==2){zi="Marti, "}
      else if(azi.getDay()==3){zi="Miercuri, "}
      else if(azi.getDay()==4){zi="Joi, "}
      else if(azi.getDay()==5){zi="Vineri, "}
      else if(azi.getDay()==6){zi="Sambata, "}
      if(azi.getMonth()==0){luna="Ianuarie "}
      else if(azi.getMonth()==1){luna="Februarie"}
      else if(azi.getMonth()==2){luna="Martie"}
      else if(azi.getMonth()==3){luna="Aprilie"}
      else if(azi.getMonth()==4){luna="Mai"}
      else if(azi.getMonth()==5){luna="Iunie"}
      else if(azi.getMonth()==6){luna="Iulie"}
      else if(azi.getMonth()==7){luna="August"}
      else if(azi.getMonth()==8){luna="Septembrie"}
      else if(azi.getMonth()==9){luna="Octombrie"}
      else if(azi.getMonth()==10){luna="Noiembrie"}
      else if(azi.getMonth()==11){luna="Decembrie"}
      data=azi.getDate()
      //-->
    </SCRIPT>
  </HEAD>
  <BODY>
    <H1 align="center">Data calendaristica</H1><HR>
    <FONT size="3" color="black" face="arial">Azi suntem in data de:
    <SCRIPT language="JavaScript">
      <!--
      <FONT size="4" color="red" face="arial">
      document.write(zi+' '+data+' '+luna)
      //-->
    </SCRIPT></FONT>
  </BODY>
</HTML>
```

Ferestre pop-up

Unul din lucrurile care nu se pot realiza folosind doar HTML este afişarea mesajelor ***pop-up***. Aceste mesaje apar într-o mică fereastră care se închide atunci când executaţi o anumită acţiune, de obicei click pe un buton, sau pe un link.

În Exemplul 13.5 este construită pagina principală din care este apelată fereastra *pop-up*.

Exemplul 13. 5

```
<HTML>
  <HEAD>
    <TITLE>javascript5</TITLE>
    <SCRIPT language="JavaScript">
      <!--
        function deschide()
        {iwin=window.open("javascript6.html", "IWIN", "status=no,
        toolbar=no, location=no, menu=no, width=200, height=200");}
```

```
      //-->
    </SCRIPT>
    </HEAD>
    <BODY>
      <H1 align="center">Fereastra pop-up</H1><HR>
      Aceasta pagina demonstreaza cum poate fi creata o fereastra pop-
      up<BR>
      Fereastra se deschide la apasarea pe legatura de mai jos si se
      inchide<BR>
      cand este apasat butonul OK<BR>
      Deschide fereastra
      <A href="#" onClick="deschide();">aici</A>.
  </BODY>
</HTML>
```

Fireşte că trebuie creat un document separat, care va fi afişat în fereastra *pop-up*. În Exemplul 13.6 este construit documentul HTML (*javascript6.html*) care va fi afişat în fereastra *pop-up* definită în exemplul anterior.

Exemplul 13. 6

```
<HTML>
  <HEAD>
    <TITLE>javascript6</TITLE>
  </HEAD>
  <BODY>
    <H3 align="center">Test pop-up</H3>
    <FONT color="green" face=arial>Ati invatat sa creati o fereastra pop-
    up<BR>
    Apasati OK ca sa reveniti in fereastra initiala
    <P>
    <FORM name="form1">
      <INPUT type="button" value="OK"
      onClick="window.close();">
    </FORM>
  </BODY>
</HTML>
```

Aspectul paginii şi al ferestrei *pop-up* este redat în Figura 13.5.

Puteţi modifica documentul HTML care se va deschide în fereastra *pop-up* după dorinţă şi îl puteţi salva sub alt nume. Nu uitaţi, însă, să faceţi modificarea corespunzătoare în funcţia `deschide()` din script.

Afişarea aleatoare a unor mesaje în pagină

Unul dintre lucrurile care îi fac pe vizitatorii paginii dumneavoastră să revină la ea cu regularitate este varietatea, fie în aspect, fie, mai ales, în conţinut.
Un mod de a face paginile mai variate este de a afişa diverse mesaje, citate sau imagini care să se schimbe de fiecare dată când este accesată pagina.

În Exemplul 13.7 este prezentat un script care realizează afişarea aleatoare a unui citat, ales dintr-un şir predefinit. Aspectul acestei pagini este redat în Figura 13.6.

Exemplul 13. 7

```
<HTML>
  <HEAD>
    <TITLE>javascript7</TITLE>
  </HEAD>
  <BODY>
```

```
    <H1 align="center">Citate</H1><HR>
    <SCRIPT language="JavaScript">
    <!--
      citat=new Array(5);
      autor=new Array(5);
      citat[0]="Omul care-si cunoaste limitele este singurul care are
      sanse sa obtina ce vrea.";
      autor[0]="Goethe";
      citat[1]="Pe stancile cele mai inalte ajung numai vulturii si
      reptilele.";
      autor[1]="Montesquieu";
      citat[2]="Putine lucruri sunt atat de greu de suportat ca un exemplu
      bun.";
      autor[2]="Mark Twain";
      citat[3]="Adevarul pur si simplu este rareori pur si niciodata
      simplu.";
      autor[3]="Oscar Wilde";
      citat[4]="Omul a inventat limbajul pentru a-si satisface nevoia
      profunda de a se plange.";
      autor[4]="Lily Tomlin";
      index=Math.floor(Math.random() * citat.length);
      document.write("<DL>\n");
      document.write("<DT>" + "\"" + citat[index] + "\"\n");
      document.write("<DD>" + "- " + autor[index] + "\n");
      document.write("</DL>\n");
    //-->
    </SCRIPT>
    <HR>
    Citatul de mai sus este generat aleator la incarcarea paginii.
  </BODY>
</HTML>
```

Dacă doriţi să inseraţi acest script în pagina dumneavoastră şi să adăugaţi şi alte citate, nu uitaţi să redimensionaţi vectorii `citat` şi `autor` la valoarea *n-1*, unde *n* este numărul de citate. Pentru a observa funcţionarea scriptului, reîncărcaţi pagina de mai multe ori.

Imagini care se schimbă (rollover images)

Un efect interesant pe care îl puteţi introduce în pagină sunt imaginile care îşi schimbă aspectul la trecerea cu mouse-ul pe suprafaţa lor. Acesta se poate obţine cu ajutorul scriptului prezentat în Exemplul 13.8. În Figura 13.7 este redat aspectul paginii descrise în acest exemplu.

Exemplul 13. 8

```
<HTML>
  <HEAD>
    <TITLE>javascript8</TITLE>
    <SCRIPT language="JavaScript">
    <!--
      if (document.images)
      {
        pic1on=new Image();
        pic1on.src="../Imagini/dovleac1.jpg";
        pic1off=new Image();
        pic1off.src="../Imagini/dovleac2.jpg";
      }
      function deschide(imgName)
      {
        if (document.images)
          {
```

```
            imgOn=eval(imgName + "on.src");
            document[imgName].src=imgOn;
            }
        }
        function inchide(imgName)
        {
          if (document.images)
            {
            imgOff=eval(imgName + "off.src");
            document[imgName].src=imgOff;
            }
        }
      //--->
      </SCRIPT>
    </HEAD>
    <BODY>
      <H1 align="center">Imagini care se schimba</H1><HR>
      <CENTER>
        <A href="#" onMouseover="deschide('pic1')"
        onMouseout="inchide('pic1')">
        <IMG src="../Imagini/dovleac2.jpg" width="100" height="100"
        border="1" name="pic1">
        </A>
      </CENTER>
    </BODY>
</HTML>
```

Validarea formularelor

Dacă exemplele prezentate până acum au avut rolul de a face pagina dumneavoastră mai atractivă, următorul exemplu este de natură să îi îmbunătăţească funcţionalitatea. O problemă cu care vă veţi confrunta fără îndoială dacă folosiţi formulare în pagină, este validarea datelor introduse de utilizator.

Vom crea o pagină foarte simplă care conţine două casete de text în care utilizatorul trebuie să introducă numele şi adresa de mail. Deoarece după completarea formularului informaţiile introduse vor fi supuse procesului de validare realizat de scriptul JavaScript, eticheta `<FORM>` va avea un conţinut diferit de cel pe care îl cunoaşteţi de la capitolul **Formulare**.
Scriptul de mai jos verifică dacă toate câmpurile de editare au fost completate. Dacă se apasă pe butonul Submit înainte de a completa ambele câmpuri, este transmis un mesaj de eroare. Datele din formular sunt transmise unui script CGI fictiv, numit *myscript.cgi* aflat în directorul *cgi-bin*.
Observaţi că atributele `action` şi `method` ale etichetei `<FORM>` pe care le cunoaşteţi de la **Formulare** sunt, de data aceasta folosite ca proprietăţi ale obiectului predefinit în JavaScript, `form`.

Documentul construit în Exemplul 13.9 conţine un formular şi scriptul care realizează validarea datelor introduse în câmpurile formularului. Din Figura 13.8 puteţi observa mesajul de eroare transmis în situaţia când unul din câmpurile formularului nu este completat.

Exemplul 13. 9

```
<HTML>
  <HEAD>
    <TITLE>javascript9</TITLE>
    <SCRIPT language="javascript">
    <!--
```

```
      function verifica(form)
      {
        if (form.nume.value== "")
          {
            alert("Va rog, introduceti numele!");
            form.nume.select();
          }
        else if (form.email.value== "")
              {
                alert("Va rog, introduceti adresa de mail!");
                form.email.select();
              }
            else
              {
                form.method="post";
                form.target="_self";
                form.action="cgi-bin/myscript.cgi";
                form.submit();
              }
      }
    //-->
    </SCRIPT>
    </HEAD>
    <BODY>
      <H1 align="center">Validarea formularelor</H1>
      <H3 align="center">(completarea tuturor campurilor)</H3><HR>
      <FORM onSubmit="verifica(this); return false;">
        Nume:
        <INPUT type="text" name="nume" size="30"><BR>
        Adresa mail:
        <INPUT type="text" name="email"><BR>
        <INPUT type="submit" name="button" value="Submit">
      </FORM>
  </BODY>
<HTML>
```

O altă verificare deosebit de utilă este dacă utilizatorul a introdus o adresă de mail validă. Scriptul din exemplul următor realizează trei verificări:

- dacă în şirul de caractere introdus există caracterul @
- dacă este prezent caracterul "**.**" urmat de trei caractere (.com, de exemplu),
- dacă este prezent caracterul "**.**" urmat de două caractere (.ro, de exemplu)

Formularul din Exemplul 13.10 conţine un câmp de editare şi un buton *Submit*. Dacă adresa de mail introdusă în câmpul de editare este incorectă se va afişa un mesaj de eroare aşa cum se vede în Figura 13.9.

Exemplul 13. 10

```
<HTML>
  <HEAD>
    <TITLE>javascript10</TITLE>
    <SCRIPT language="javascript">
    <!--
      function verifica_mail(form)
      {
        verif=form.email.value
        if ((verif.indexOf('@') < 0) || ((verif.charAt(verif.length-4) !
        ='.') && (verif.charAt(verif.length-3) !='.')))
          {
            alert("Nu ati introdus o adresa de mail valida. Va rog,
            incercati din nou!");
```

```
            form.email.select();
            return false;
          }
          else
            {
              form.method="post";
              form.target="_self";
              form.action="cgi-bin/myscript.cgi";
              form.submit();
            }
      }
    //-->
    </SCRIPT>
  </HEAD>
  <BODY>
    <H1 align="center">Validarea formularelor</H1>
    <H3 align="center">(adresa de mail corecta)</H3><HR>
    <FORM onSubmit="verifica_mail(this);return false;">
      Adresa mail:
      <INPUT type="text" name="email"><BR>
      <INPUT type="submit" name="button" value="Submit">
    </FORM>
  </BODY>
<HTML>
```

Puteţi folosi în paginile dumneavoastră oricare dintre aceste scripturi, desigur, făcând modificările necesare legate de structura site-ului dumneavoastră. De exemplu, în scripturile care realizează validarea formularelor, pentru ca acestea să fie funcţionale, trebuie să precizaţi localizarea corectă a scriptului CGI care face prelucrarea lor.

Din exemplele prezentate probabil că v-aţi putut crea o imagine despre modul în care pot fi utilizate scripturile JavaScript într-o pagină Web. Dacă sunteţi entuziasmat încă de pe acum, veţi fi şi mai mult dacă veţi depune puţin efort pentru a învăţa limbajul şi veţi putea crea scripturi mult mai complexe decât acestea.

Aveţi însă grijă la aspectele legate de viteza de încărcare a paginii: aplicaţiile complexe şi de mari dimensiuni încetinesc încărcarea paginii unde sunt folosite. Ca şi în cazul imaginilor, nu folosiţi scripturi de care nu este nevoie, doar pentru a arata cu orice preţ că sunteţi un bun programator. Una dintre regulile de aur ale unei pagini Web bune, regulă atât de adesea încălcată, este simplitatea.

5. Rezumat

JavaScript este un limbaj de programare orientat pe obiecte şi evenimente care permite extinderea capacităţilor limbajului HTML.

Scripturile JavaScript sunt incluse într-un document HTML prin intermediul etichetei container `<SCRIPT>` sau prin intermediul procedurilor eveniment. Pentru a evita afişarea în pagină a codului scriptului de browserele care nu suportă scripturi, se obişnuieşte ca scriptul să fie inclus între etichetele de comentariu.

Scripturile pot fi inserate direct în documentul HTML sau pot fi stocate în fişiere externe care sunt apelate în documentul respectiv.

Cu ajutorul scripturilor JavaScript se pot realiza sarcini diverse cum ar fi crearea ferestrelor pop-up, afişarea datei curente în pagină, validarea formularelor şi altele.

În capitolul următor vom discuta despre o altă metodă care îmbogăţeşte limbajul HTML cu noi posibilităţi de a controla aspectul paginilor, şi anume programarea cu ajutorul foilor de stiluri (*Style Sheets*) Capitolul 14 Foi de stiluri (CSS)

Foile de stiluri HTML (*Cascading Style Sheets*) reprezintă o inovaţie în dezvoltarea World Wide Web, în ciuda faptului că ideea grupării elementelor de formatare a documentelor a apărut ceva mai demult. Aplicarea stilurilor reprezintă o extindere importantă a posibilităţilor de design, evitând utilizarea de fişiere grafice mari ce determină încetinirea încărcării paginilor şi manipularea lor greoaie.

Folosind stilurile HTML puteţi fi sigur că cititorii vor vedea textul din pagină exact aşa cum a fost el proiectat. Textul şi proprietăţile acestuia, care au fost iniţial controlate de browsere, se reîntorc la autor, acolo unde le este, de fapt, locul.

1. Ce este un stil?

Un **stil** reprezintă o colecţie de atribute ale textului şi ale modului de aranjare a documentului care pot fi aplicate în mod selectiv unui document sau doar unei părţi din acesta. Aceste atribute pot fi tipul de font, mărimea şi grosimea acestuia, marginile, paragrafele şi orice altceva ce poate influenţa aspectul textului în pagină. Gruparea lor în stiluri permite autorului să aplice aceeaşi colecţie de atribute la diferite părţi ale unui document.

Stilurile aplicate unui document HTML au multe avantaje pentru creatorii de pagini Web:

- Oferă control crescut asupra aspectului şi plasării textului în pagină
- Reduc "învălmăşeala" produsă de multitudinea de deschideri şi închideri ale etichetelor care descriu elementele individuale ale textului
- Procesul de modificare a diferitelor elemente din pagină se simplifică

De exemplu, dacă doriţi ca titlurile dumneavoastră să aibă un alt tip de font decât textul obişnuit, să fie de dimensiune mai mare şi scrise cu caractere îngroşate şi italice, ar trebui să definiţi aceste atribute de formatare pentru fiecare titlu în parte. Folosind stilurile nu aveţi nevoie decât să creaţi o singură definiţie de stil care să conţină atributele de formatare dorite, pe care să o aplicaţi la fiecare titlu.

În plus, folosirea stilurilor reduce considerabil efortul depus atunci când doriţi să aduceţi modificări aspectului şi aranjării elementelor din paginile dumneavoastră. În loc să parcurgeţi fiecare document în parte şi să faceţi modificări asupra fiecărui element, nu mai este necesar să operaţi modificări decât asupra foii de stiluri care controlează aceste elemente.

2. Tipuri de foi de stiluri

Stilurile HTML pot fi aplicate într-o pagină Web în trei moduri:

Încapsulate (*embedded*): stilurile sunt incluse în documentul asupra căruia se aplică, şi anume în secţiunea <HEAD> a documentului. Prin includerea lor în antetul documentului, stilurile rămân invizibile pentru vizitatorul paginii.

Legate (*linked*): stilurile sunt definite în fişiere separate de documentul HTML. Documentul face apel la foaia de stiluri prin intermediul etichetei <LINK>. Folosirea acestui tip de stiluri face posibilă utilizarea aceleiaşi foi de stil pentru documente diferite. De asemenea, este posibilă aplicarea mai multor foi de stiluri pentru acelaşi document.

In-line: stilurile sunt incluse ca atribute în cadrul etichetelor HTML din document. Aceasta înseamnă că ele vor afecta doar elementul asupra căruia sunt aplicate. Este o modalitate mai puţin utilizată deoarece contrazice principiul general al stilurilor, acela de a simplifica şi de a face mai lizibil codul documentului HTML.

Sintaxa definiţiilor de stil este, în general, aceeaşi, indiferent de modul cum sunt ele aplicate. Cele trei tipuri de stiluri pot fi, de asemenea, combinate în cadrul aceluiaşi document. Relaţiile dintre diferitele tipuri de stiluri realizează efectul de cascadă care dă numele acestei metode.

Stiluri încapsulate

Crearea unui astfel de stil se realizează folosind eticheta **<STYLE> </STYLE>.** Eticheta <STYLE> este o etichetă container, deci este obligatorie prezenţa etichetei de închidere </STYLE>. Între cele două etichete se introduc definiţiile de stil. Eticheta de stil este plasată în antetul documentului adică în secţiunea <HEAD>.

În Exemplul 14.1 este prezentată o definiţie de stil care realizează afişarea tuturor titlurilor de nivel 1 (adică a textelor cuprinse între etichetele <H1> </H1>) cu caractere bold şi de culoare verde. În plus, textele incluse între etichetele <P> </P> vor fi afişate cu fonturi arial, de mărime 2 şi culoare violet.
De asemenea, este creat un stil "normal" care poate fi aplicat asupra oricărui text. Prin intermediul său, textul este afişat cu caractere mai mari şi de culoare oranj. Aspectul paginii în care este inclusă această definiţie de stil este cel din Figura 14.1.

Exemplul 14. 1

```
<HTML>
  <HEAD>
    <TITLE>stiluri1</TITLE>
    <STYLE>
    <!--
      H1 {color: #008000; font-weight: bold}
      P {font-family: Arial; color: #800080; font-size: 14px}
      .normal {font-size: large; color: #FF8000}
    -->
    </STYLE>
  </HEAD>
  <BODY>
    <H1 align="center">Stiluri</H1><HR>
    <P>Textul din acest paragraf este formatat cu ajutorul stilurilor</P>
    Acesta este un text neformatat
    <H1>Titlu</H1>
    <SPAN class="normal">Acesta este stilul normal</SPAN>
```

```
  </BODY>
</HTML>
```

Observaţi că definiţia stilurilor a fost plasată între etichetele HTML de comentariu, în aceeaşi manieră ca şi la scripturile JavaScript. Motivul este acelaşi: dacă browserul cu care este vizualizată pagina nu suportă foi de stiluri, se evită astfel afişarea în pagină a codului HTML.

Stiluri legate – foi de stiluri externe

Stilurile incluse în pagină se aplică elementelor prezente în respectivul document HTML, reducând dimensiunea codului şi efortul de a defini fiecare element în parte. Această idee se poate extinde la nivelul mai multor documente care pot beneficia, toate, de aceleaşi stiluri, reunite într-o foaie de stiluri externă.

Avantajul folosirii foilor de stiluri externe este dublu. Pe de-o parte ele se pot aplica la nivelul mai multor documente HTML, realizând astfel o legătură de stil între ele, lucru deosebit de util la construirea unui site. Pe de altă parte, acelaşi document poate folosi foi de stiluri diferite, oferind vizitatorului posibilitatea de a opta pentru unul sau altul dintre ele, în funcţie de propriile preferinţe.

O foaie de stiluri este un fişier text care conţine regulile de stil definite în aceeaşi manieră folosită la stilurile incluse în pagină. Odată creată o foaie de stiluri, ea trebuie salvată cu extensia ***.css***. Foile de stiluri pot fi stocate într-un folder separat sau în acelaşi folder în care sunt plasate şi documentele HTML.

Apelul foilor de stiluri se poate realiza în două moduri:

- folosind eticheta **<LINK>**
- folosind funcţia **@import**

Cel mai folosit mod de apelare a unei foi de stiluri este cu ajutorul etichetei `<LINK>` conform următoarei sintaxe:

```
<LINK rel="stylesheet" href="nume_foaie_stiluri.css">
```

În Exemplul 14.2 este construită o foaie de stiluri externă care va fi apelată cu ajutorul etichetei `<LINK>`. Foaia de stiluri a fost salvată sub numele *stiluri2.css*

Exemplul 14. 2

```
BODY
{background-color: #CCFFCC;
font-family: Arial, sans-serif;
color: #330066;
padding: 50px, 70px}
A:link {color: #CC9900}
A:visited {color: #660000}
A:hover {color: #FFCC00}
A:active {color: #FF0000}
H1 {color: #996633;
background-color: #FFFFCC}
```

Documentul HTML care urmează apelează foaia de stiluri de mai sus este prezentat în Exemplul 14.3. După cum observaţi, foaia *stiluri2.css* a fost apelată în antetul documentului, prin intermediul etichetei `<LINK>` cu atributul `rel="stylesheet"`. Atributul `href` al etichetei are ca valoare numele (şi adresa relativă, dacă este necesar) al foii de stiluri apelate. În documentul HTML am

inclus şi o legătură, pentru a exemplifica modul în care foaia de stiluri schimbă culorile legăturii. Atributul **hover** se referă la proprietatea ca legătura să îşi schimbe culoarea la trecerea cu mouse-ul peste ea, fără a face click. Pagina descrisă în acest exemplul are aspectul din Figura 14.2.

Exemplul 14. 3

```
<HTML>
  <HEAD>
    <TITLE>stiluri3</TITLE>
    <LINK rel="stylesheet" href="stiluri2.css">
  </HEAD>
  <BODY>
    <H1 align="center">Foi de stiluri externe</H1>
    Acest exemplu ilustreaza modul in care este inclusa in pagina o foaie
    de stiluri externa
    <P>Mai multe exemple in <A href="anexa1.html">anexa2</A>
  </BODY>
</HTML>
```

Stiluri in-line

Spre deosebire de stilurile încapsulate şi de foile de stiluri externe, stilurile *in-line* fac parte chiar din corpul documentului HTML. Ele se aplică prin folosirea atributului **style** în asociere cu etichetele HTML standard. Reţineţi, deci, că `style` poate fi atât etichetă în sine cât şi atribut al altor etichete.

Definiţiile de stil *in-line* se aplică numai asupra elementelor incluse între etichetele care au asociat atributul `style`. Din acest motiv, dacă dorim să repetăm în alt loc din cuprinsul documentului aceleaşi definiţii de stil, ele vor trebui repetate, încărcând astfel documentul HTML. Totuşi, utilitatea stilurilor *in-line* este aceea că fiind definite chiar în cuprinsul documentului, definiţiile lor sunt mai puternice decât cele din stilurile încapsulate sau externe. Dacă, de exemplu, am creat o foaie de stiluri pe care dorim să o aplicăm unui document, şi dacă în cadrul acestui document avem un anumit element particular căruia dorim să-i dăm un alt aspect decât cel prevăzut în foaia externă, putem aplica acelui element un stil *in-line* care se va referi strict la el.

Iată un exemplu de aplicare a unui stil *in-line* asociat etichetei `<P>`:

```
<P style="color: red; font-family: arial; font-weight: bold">Textul din
acest paragraf este scris cu fonturi arial, ingrosate, de culoare rosu</P>
```

Observaţi că la stilurile *in-line* definiţiile de stil sunt incluse între apostrofuri şi nu între acolade, ca la celelalte tipuri de stiluri.

Atunci când lucraţi cu documente HTML deja existente cărora doriţi să le aplicaţi stiluri *in-line*, este recomandat să folosiţi etichetele **<DIV>** şi **<SPAN>**. Acestea vă permit să aplicaţi stilurile fără a afecta codul HTML deja existent sau aspectul paginii în browserele care nu suportă stiluri.

Exemplul 14.4 ilustrează folosirea etichetei `<SPAN>` aspectul paginii fiind redat în Figura 14.3.

Exemplul 14. 4

```
<HTML>
  <HEAD>
    <TITLE>stiluri4</TITLE>
  </HEAD>
  <BODY>
```

```
    <H1 style="color:red;font-famyly:arial">Stiluri in-line</H1><HR>
    <P>Textul din acest paragraf este divizat folosind eticheta span</P>
    <SPAN style="font-family:arial; color: blue">Text scris cu albastru si
    fonturi arial</SPAN><BR>
    <SPAN style="font-family: courier; size: medium; color:green">Text
    scris cu verde si fonturi courier medium</SPAN>
  </BODY>
</HTML>
```

La fel ca şi la capitolele anterioare, nu vom intra în amănunte privind programarea cu ajutorul stilurilor. Deşi stilurile sunt intuitive şi uşor de aplicat, totuşi o abordare exhaustivă a acestui subiect depăşeşte obiectul acestei cărţi. Prezentăm, totuşi, în continuare, câteva exemple care vă pot fi utile în paginile dumneavoastră.

3. Efecte obţinute cu ajutorul stilurilor

Aspectul textului

Iată câteva atribute care se pot asocia cu diverse etichete HTML permiţând schimbarea aspectului textului.

Tabel 14. 1

Atributul	Efectul
`color`	Setează culoarea textului.
`background-color`	Setează culoarea fundalului unui obiect.
`font-family`	Setează tipul de font.
`font-size`	Setează dimensiunea fonturilor. Valorile pot fi exprimate în pixeli (px) sau puncte (pt).
`font-weight`	Setează grosimea fonturilor. Valorile posibile pot fi: normal, bold, bolder, lighter.
`font-style`	Setează stilul de font. Valorile pot fi: normal, italic, oblique.
`letter-spacing`	Setează spaţierea literelor. Valorile pot fi: pixeli (px), puncte (pt), inci (in), centimetri (cm) sau milimetri (mm).
`word-spacing`	Setează spaţierea cuvintelor. Valorile pot fi: pixeli (px), puncte (pt), inci (in), centimetri (cm) sau milimetri.
`text-transform`	Valorile posibile pot fi: capitalize, uppercase, lowercase, none.
`text-decoration`	Valorile pot fi: underline, overline, line-through, blink.
`hover`	Schimbă aspectul unei legături când mouse-ul se află deasupra ei.
`link`	Defineşte aspectul unei legături.
`active`	Defineşte aspectul legăturii active.
`visited`	Defineşte aspectul legăturii vizitate.
`first-letter`	Schimbă aspectul primei litere dintr-un cuvânt.
`first-line`	Schimbă aspectul primei linii dintr-un text.

În exemplele următoare sunt ilustrate câteva din aceste atribute.
Exemplul 14.5 include o definiţie de stil în care sunt definiţi aşa numiţii **selectori de clasă**, în acest exemplu `B.titlu` şi `B.subtitlu`. În corpul documentului apelul acestora a fost făcut prin construcţia:
`<B class="titlu">Titlu</B>`
Atributul **class** care se poate asocia etichetei <B> ca în acest exemplu sau altor etichete de formatare a textului are ca valoare un identificator care a fost definit în prealabil în cadrul definiţiei de stil.
Aspectul paginii descrise în acest exemplu este cel din Figura 14.4.

Exemplul 14. 5

```
<HTML>
  <HEAD>
    <TITLE>stiluri5</TITLE>
    <STYLE>
    <!--
      B.titlu {font-size: 20 pt; font-weight: bolder; letter-spacing:5px}
      B.subtitlu {font/size: 15 pt; letter-spacing: 5px}
    -->
    </STYLE>
  </HEAD>
  <BODY>
    <H1 align="center">Aspectul textului</H1><HR>
    <P>
    Text normal<BR>
    <B>Text ingrosat</B><BR>
    <B class="titlu">Titlu</B><BR>
    <B class="subtitlu">Subtitlu</B>
    </P>
  </BODY>
</HTML>
```

Exemplul 14.6 foloseşte stiluri la formatarea titlului de nivel H1 şi a textului din cadrul paragrafului, folosind atributul `first-letter` care schimbă aspectul primei litere din paragraf. Şi în acest exemplu a fost folosit un selector de clasă şi anume `P.primalit`.
Aspectul paginii care foloseşte aceste stiluri este redat în Figura 14.5.

Exemplul 14. 6

```
<HTML>
  <HEAD>
  <TITLE>stiluri6</TITLE>
  <STYLE>
  <!--
    P.primalit:first-letter {font-weight: bolder; font-style: italic;
    font-size:50 pt; color:red}
    H1 {color:blue; letter-spacing:5 px}
  -->
  </STYLE>
  </HEAD>
  <BODY>
    <H1 align="center">Efecte asupra textului</H1><HR>
    <P class="primalit">Textul din acest paragraf foloseste atributul
    first-letter</P>
  </BODY>
</HTML>
```

Bare de derulare colorate

Unul dintre efectele des întâlnite în paginile Web este prezenţa barelor de derulare colorate. Acest efect se poate, de asemenea, obţine prin aplicarea stilurilor. Pentru a defini culoarea barelor de derulare se folosesc câteva atribute, pe care le definim mai jos.

Tabel 14. 2

Atributul	Efectul
`scrollbar-base-color`	Defineşte culoarea de bază a barei, a butoanelor cu săgeţi şi a fundalului pe care se deplasează bara. Dacă doriţi să realizaţi rapid o bară de derulare nu este necesar să setaţi decât acest atribut şi pe cel următor. Setarea celorlalte atribute se suprapune şi anulează efectele acestuia.
`scrollbar-arrow-color`	Defineşte culoarea săgeţilor de pe bară
`scrollbar-face-color`	Defineşte culoarea suprafeţei barei de derulare şi a butoanelor pe care sunt săgeţile. De asemenea, stabileşte culoarea fundalului pe care se deplasează bara.
`scrollbar-highlight-color`	Defineşte culoarea părţilor "iluminate" ale barei şi butoanelor cu săgeţi (marginile de sus şi din stânga) şi a fundalului pe care se deplasează bara
`scrollbar-darkshadow-color`	Defineşte culoarea părţilor "întunecate" ale barei şi butoanelor cu săgeţi (marginile de jos şi din dreapta)
`scrollbar-shadow-color`	Defineşte culoarea marginilor de jos şi din dreapta ale barei şi butoanelor cu săgeţi
`scrollbar-3dlight-color`	Defineşte culoarea marginilor de sus şi din stânga ale barei şi butoanelor cu săgeţi
`scrollbar-track-color`	Defineşte culoarea fundalului barei

După cum observaţi, unele dintre aceste atribute îşi suprapun efectele. Există anumite reguli care stabilesc prioritătile atributelor pe care nu le vom aborda aici. Ca începător, este recomandat să nu folosiţi atribute care se suprapun pentru a nu obţine rezultate neconforme cu intenţiile dumneavoastră. Exemplul următor ilustrează modul de construire a barelor de derulare colorate.

În Exemplul 14.7 am construit o foaie de stiluri externă salvată sub numele *stiluri7.css.*

Exemplul 14. 7

```
BODY
{
scrollbar-face-color:blue;
```

```
scrollbar-arrow-color:yellow;
scrollbar-shadow-color:red;
scrollbar-3dlight-color:yellow;
scrollbar-track-color:cyan;
}
```

Documentul HTML care apelează foaia de stiluri este construit în Exemplul 14.8, pagina descrisă de acest document având aspectul din Figura 14.6.

Exemplul 14. 8

```
<HTML>
  <HEAD>
    <TITLE>stiluri8</TITLE>
    <LINK rel="stylesheet" href="stiluri7.css">
  </HEAD>
  <BODY>
    <H1 align="center">Bare de derulare colorate</H1>
    a<BR>a<BR>a<BR>a<BR>a<BR>a<BR>a<BR>
    a<BR>a<BR>a<BR>a<BR>a<BR>a<BR>a<BR>
    a<BR>a<BR>a<BR>a<BR>a<BR>a<BR>a<BR>
  </BODY>
</HTML>
```

Desigur, culorile pe care le alegeţi pentru bara de derulare trebuie selectate cu grijă, astfel încât să se armonizeze cu restul paginii. Folosirea unei foi de stiluri externe este utilă deoarece puteţi stabili astfel culoarea barei pentru toate paginile site-ului dumneavoastră.

Browserele Netscape nu suportă schimbarea culorilor barei de derulare.

Formulare colorate

Exemplul care urmează (Exemplul 14.9) foloseşte stiluri *in-line* pentru a crea formulare colorate care pot da paginii un aspect mai atractiv. Am folosit pentru setarea culorilor codurile hexazecimale pe care le puteţi găsi în **Anexa 3**. Ca şi la barele de derulare, culorile formularelor trebuie alese cu grijă pentru a nu da paginii un aspect încărcat şi lipsit de bun-gust. Pagina descrisă în acest document arată ca în Figura 14.7.

Exemplul 14. 9

```
<HTML>
  <HEAD>
    <TITLE>stiluri9</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Formulare colorate</H1><HR>
    <FORM>
      <INPUT name="text" style="background-color: #00CCFF; color: #FFFFFF;
      font-weight: bold" value="Salut!">
      <P>
      <TEXTAREA rows="5" cols="20" style="color: #0066CC; background-
      color: #CCCCCC; scrollbar-base-color:red;">Care sunt impresiile
      tale?</TEXTAREA>
      <P>
      <INPUT type="submit" value="trimite" style="background-color:
      #0066CC">
    </FORM>
  </BODY>
</HTML>
```

Vizualizate în Netscape formularele din exemplul de mai sus vor avea aspectul normal, necolorat.

4. Rezumat

Foile de stiluri reprezintă un instrument important în elaborarea documentelor HTML, mai ales dacă dorim să construim un site Web. Acestea permit păstrarea unui aspect unitar şi coerent pe tot parcursul site-ului.

Un stil este o colecţie de atribute ale textului şi ale modului de aranjare a documentului care pot fi aplicate în mod selectiv unui document sau doar unei părţi din acesta.

Stilurile se pot aplica unui document în trei moduri:

- încapsulate - prin intermediul etichetei `<STYLE>` plasată în antetul documentului
- legate - prin intermediul etichetei `<LINK>` plasată de asemenea în antetul documentului şi care face legătura cu foaia de stiluri externă
- *in-line* - prin intermediul atributului `style` asociat cu diverse etichete HTML.

Cu ajutorul stilurilor se pot obţine efecte deosebite în cadrul documentelor HTML: se poate modifica aspectul textului, se pot construi bare de derulare colorate, formulare colorate şi altele.

În capitolul următor vom discuta despre câteva elemente de grafică avansată prin intermediul cărora pagina dumneavoastră poate deveni mai interesantă şi mai atractivă.

Capitolul 15
Elemente de grafică avansată

1. Imagini hartă

În mod obişnuit, când folosim o imagine drept legătură, includem eticheta `<IMG>` în interiorul etichetei pentru link-uri , `<A>`. În continuare este prezentată o altă modalitate de a folosi o imagine ca legătură şi anume crearea unei imagini hartă.

Ce este o imagine hartă?

O **imagine hartă** este o zonă activă pe care se poate executa click cu mouse-ul, apăsarea diferitelor regiuni ale imaginii ducând la deschiderea unor documente HTML diferite. Cu alte cuvinte, o imagine hartă este o imagine împărţită în regiuni, iar fiecare regiune reprezintă o legătură către un alt document. Un exemplu de utilizare a imaginilor hartă îl constituie barele de butoane folosite în multe site-uri ca instrumente de navigare.

Imaginile hartă oferă posibilitatea de a folosi o singură imagine pentru a furniza legături către mai multe pagini. Vă puteţi imagina o imagine hartă ca fiind compusă din două părţi: imaginea în sine şi o hartă invizibilă care este aplicată peste imagine, împărţind-o în regiuni. Această hartă foloseşte pentru definirea regiunilor forme prestabilite: poligoane, cercuri sau dreptunghiuri. În general,

imaginile care se pretează la a fi folosite ca imagini hartă sunt cele care conţin forme geometrice care conduc la o împărţire naturală a imaginii în regiuni.

În funcţie de modul cum sunt prelucrate imaginile şi identificate adresele spre care indică fiecare regiune, imaginile hartă pot fi de două tipuri:
- imagini de tip server
- imagini de tip client

Imagini hartă pentru server (server side)

O imagine hartă de **tip server** se caracterizează prin faptul că face apel la o aplicaţie (un script) de pe server care realizează determinarea regiunii pe care s-a făcut click şi încarcă documentul HTML asociat acelei regiuni. Nu este nevoie să vă faceţi griji în legătură cu scriptul care prelucrează imaginea hartă deoarece majoritatea serverelor au deja instalată o astfel de aplicaţie, cele mai cunoscute fiind Imagemap şi HTImage.

Folosirea imaginilor hartă de tip server parcurge următorii paşi:

- vizitatorul face click pe o anumită zonă a imaginii
- coordonatele acestei zone sunt transmise serverului
- scriptul aflat pe server asociază aceste coordonate cu URL-ul unui anumit document HTML (lista adreselor URL ale imaginilor asociate cu regiunile hărţii se află plasată tot pe server)
- adresa URL împreună cu coordonatele regiunii sunt trimise înapoi la browser care deschide documentul HTML corespunzător.

Este lesne de observat că parcurgerea tuturor acestor etape precum şi interacţiunea cu serverul poate încetini mult procesul de încărcare a documentelor HTML asociate regiunilor hărţii. Singurul avantaj al folosirii imaginilor hartă este faptul că funcţionează în toate browserele. Totuşi, deoarece versiunile mai noi decât Netscape 2.02 şi Internet Explorer 2.0 recunosc imaginile hartă de tip client, se va renunţa treptat la folosirea imaginilor hartă de tip server.

Imagini hartă de tip client (client side)

Imaginile hartă de **tip client** lucrează independent de server. Când folosim acest tip de imagini hartă plasăm toate datele necesare prelucrării hărţii chiar în corpul documentului HTML astfel încât browserul îl execută fără a mai fi necesară o intervenţie din partea serverului.

Imaginile hartă de tip client au avantaje evidente: sunt mult mai rapide, funcţionarea lor este mult simplificată, şi, în plus, la trecerea cu mouse-ul peste hartă, în bara de status a ferestrei browserului apare adresa URL asociată fiecărei regiuni, ceea ce oferă informaţii suplimentare vizitatorului.

Crearea unei imagini hartă de tip client

Pentru a include în documentul HTML o hartă pentru o anumită imagine se foloseşte eticheta container **<MAP> </MAP>**. Între aceste etichete se plasează mai multe etichete **<AREA>** care au rolul de a defini coordonatele şi forma regiunilor pe care dorim să le delimităm pe imagine, ca în exemplul următor:

```
<MAP name="harta">
  <AREA shape="rect" coords="x1, y1, x2, y2" href="URL1">
  <AREA shape="poly" coords="x1, y1, x2, y2, …, xn, yn" href="URL2">
```

```
  <AREA shape="circle" coords="x, y, raza" href="URL3">
</MAP>
```

Regiunile definite în acest exemplu sunt un dreptunghi, pentru care sunt specificate coordonatele vârfurilor din stânga sus şi dreapta jos, un poligon cu n vârfuri, definit prin coordonatele lor, şi un cerc definit prin coordonatele centrului şi rază.

Pentru a comunica browserului ce imagine trebuie să folosească pentru a crea imaginea hartă se foloseşte eticheta pentru imagini **<IMG>**, atributul **src** având ca valoare adresa fişierului grafic folosit. În plus, în cadrul etichetei <IMG> se foloseşte atributul **usemap** care are rolul de ancoră (legătură internă). De exemplu, dacă harta pe care o folosim a fost definită în eticheta <MAP> cu numele "harta1" şi foloseşte fişierul *imagine1.gif* atunci eticheta <IMG> va avea următoarea structură:

```
<IMG src="imagine1.gif" usemap="#harta1" width="100" height="100"
border="0">
```

În exemplul următor (Exemplul 15.1) vom crea o imagine hartă formată din patru regiuni, trei dreptunghiuri şi un poligon cu patru vârfuri. Pentru a determina coordonatele vârfurilor fiecărei regiuni am folosit Microsoft Photo Editor, dar puteţi folosi orice alt editor grafic. La plasarea cursorului mouse-ului oriunde în cadrul imaginii, în bara de status (colţul din stânga jos) veţi observa coordonatele în pixeli, ale punctului în care vă găsiţi. Am asociat fiecărei regiuni câte un document HTML dintre cele create la capitolele anterioare. Aspectul acestei pagini este redat în Figura 15.1.

Exemplul 15. 1

```
<HTML>
  <HEAD>
    <TITLE>grafica1</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Imagini harta</H1><HR>
    <MAP name="harta">
      <AREA shape="poly" coords="29, 109, 44, 122, 111, 24, 125, 36"
      href="culori1.html">
      <AREA shape="rect" coords="132, 35, 245, 57" href="fonturi1.html">
      <AREA shape="rect" coords="114, 61, 224, 92" href="liste1.html">
      <AREA shape="rect" coords="81, 95, 227, 125" href="stiluri1.html">
    </MAP>
    <H3>Pentru a deschide cartile din imagine faceti click pe una dintre
    ele</H3>
    <IMG src="../Imagini/books1.gif" usemap="#harta" width="283"
    height="142" border="0">
  </BODY>
</HTML>
```

2. Imagini animate

O **imagine animată** este formată dintr-o serie de imagini (pentru animaţie se foloseşte formatul GIF) care sunt afişate una după cealaltă creând aparenţa unei mişcări continue. Există multe site-uri care oferă imagini animate pe care le puteţi descărca şi salva pe hard-disk pentru a le folosi în cadrul site-ului dumneavoastră. Este suficient să deschideţi motorul de căutare pe care îl folosiţi de obicei şi să introduceţi drept cheie de căutare cuvintele "animated gifs" şi veţi obţine un foarte mare număr de rezultate.

Aveţi, însă, şi posibilitatea de a vă crea propriile imagini animate. Acest proces poate fi destul de obositor, deoarece trebuie să creaţi toate imaginile care urmează să facă parte din procesul de animaţie. În continuare este prezentat modul de realizare al unei animaţii foarte simple, care realizează afişarea literă cu literă a cuvântului ANIMAŢIE.

Fiecare secvenţa din cuvânt reprezintă o imagine distinctă, astfel că vom crea 9 fişiere GIF care vor conţine secvenţele: ANIMATIE, ANIMATI, ANIMAT, ANIMA, ANIM, ANI, AN, A. Ultimul fişier va conţine o imagine fără nici o literă.

Deoarece imaginile sunt foarte simple am folosit pentru crearea lor utilitarul *Paint*, din *Windows > Accessories*, dar dacă doriţi să creaţi imagini mai complexe este necesar să utilizaţi un editor grafic mai performant.

După crearea imaginilor şi salvarea fişierelor cu extensia *.gif* se trece la ce-a de-a doua etapă: reunirea lor pentru a crea procesul de animaţie. Această animaţie a fost realizată cu ajutorul aplicaţiei numite GIF Construction Set pe care o puteţi găsi la adresa: http://www.mindworkshop.com/alchemy/
Programul oferă, pe lângă crearea animaţiei, şi multe alte opţiuni: crearea de butoane, bannere, efecte de tranziţie ale paginilor, etc.

În meniul *File* al programului există un *Animation Wizard* care vă va conduce pas cu pas de-a lungul procesul de creare a animaţiei. Wizard-ul vă va cere câteva informaţii printre care ce fişiere doriţi să facă parte din procesul de animaţie (în cazul nostru cele nouă fişiere GIF), dacă doriţi ca procesul să se desfăşoare fără oprire şi care este intervalul de timp dintre două afişări. După ce *Animation Wizard* a definitivat procesul de animaţie nu mai aveţi altceva de făcut decât să salvaţi fişierul astfel obţinut şi să-l folosiţi în pagina dumneavoastră. Includerea fişierului care conţine imaginea animată într-un document HTML se face ca şi la o imagine obişnuită folosind eticheta `<IMG>`.

Exemplul 15.2 ilustrează modul cum este inclusă în pagină o imagine animată aspectul paginii fiind cel din Figura 15.2.

Exemplul 15. 2

```
<HTML>
  <HEAD>
    <TITLE>grafica2</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">Imagini animate</H1><HR>
    <P><IMG src="../Imagini/animatie.gif">
  </BODY>
</HTML>
```

Deşi imaginile animate sunt atractive şi dau paginii un aspect dinamic, totuşi excesul de animaţie poate avea efecte contrare. Prea multe imagini animate sau o animaţie prea complexă pot conduce la mărirea timpului de încărcare a paginii. În plus, animaţia poate deveni obositoare şi poate distrage atenţia vizitatorului de la conţinutul paginii. Vă recomandăm, deci, să folosiţi imaginile animate cu grijă şi numai atunci când au un rol bine definit în cadrul paginii.

3. Imagini transparente

Adesea o imagine inclusă într-o pagină arată mai bine dacă are un fundal transparent. Un fundal transparent înseamnă că deşi imaginea are un fundal de o anumită culoare, el nu este vizibil în pagină deoarece fundalul paginii înlocuieşte fundalul imaginii. Desigur, orice culoare din imagine poate fi făcută

transparentă, dar cel mai adesea această tehnică se aplică asupra culorii de fundal.

Dacă dorim să realizăm un fundal transparent pentru o imagine trebuie să ţinem seama de o serie de lucruri:

- Fundalul trebuie să aibă o singură culoare şi nu o combinaţie de culori.
- Culoarea de fundal nu trebuie să mai fie prezentă altundeva în cadrul imaginii, deoarece transparenţa se va aplica tuturor zonelor colorate cu aceeaşi culoare
- Este recomandat ca fundalul paginii de Web să aibă atributul `bgcolor` setat în mod explicit
- Fotografiile sau imaginile cu mai mult de 256 de culori nu sunt potrivite pentru a fi transformate în imagini transparente deoarece în cadrul lor este greu de definit o singură culoare.
- Imaginile pe care dorim să le transformăm în imagini transparente trebuie să fie în format GIF

În Exemplul 15.3 am folosit o imagine cu fundalul de culoare albă pe care am prelucrat-o cu ajutorul aplicaţiei Microsoft Photo Editor. Nu este necesar decât să deschideţi fişierul GIF cu imaginea dorită, să selectaţi din meniul *Tools* opţiunea *Set Transparent Color* care permite alegerea şi setarea culorii pe care dorim să o facem transparentă, apoi să salvaţi noua imagine sub un alt nume decât cea originală. Indiferent care este culoarea de fundal a paginii dumneavoastră, fundalul imaginii transparente va avea aceeaşi culoare cu pagina aşa cum se observă din Figura 15.3.

Exemplul 15. 3

```
<HTML>
  <HEAD>
    <TITLE>grafica3</TITLE>
  </HEAD>
  <BODY bgcolor="cyan">
    <H1 align="center">Imagini transparente</H1><HR>
    <DIV align="center">
      <H4>Imaginea originala</H4>
      <IMG src="../Imagini/gold.gif">
      <H4>Imaginea transparenta</H4>
      <IMG src="../Imagini/gold_tr.gif">
    </DIV>
  </BODY>
</HTML>
```

Pentru a verifica modul în care funcţionează imaginea transparentă pe diferite culori ale fundalului paginii, atribuiţi atributului `bgcolor` şi alte valori decât cea din exemplu.

4. Rezumat

În documentele Web pot fi incluse diverse elemente avansate de grafică:
- Imagini hartă
- Animaţie
- Imagini transparente

O imagine hartă este o imagine care conţine zone active pe care se poate efectua click cu mouse-ul, fiecare zonă conducând la deschiderea unui document HTML diferit.

Imaginile hartă sunt de două tipuri:
- Imagini hartă de tip server

- Imagini hartă de tip client (cele mai utilizate)

O imagine hartă de tip client se inserează într-un document cu ajutorul etichetei container <MAP> </MAP>. În cadrul acestei etichete fiecare zonă este definită prin intermediul etichetei <AREA>.

Imaginile animate sunt fişiere GIF care se includ în pagină ca orice altă imagine, folosind eticheta <IMG>. Pentru a crea o imagine animată aveţi nevoie de un editor grafic, cu ajutorul căruia să creaţi imaginile care vor face parte din componenţa animaţiei şi de un program special care să le reunească într-un singur fişier GIF.

Imaginile transparente sunt imagini la care una dintre culori (cel mai adesea culoarea de fundal) este transformată în culoare transparentă, astfel încât la includerea ei într-o pagină Web fundalul imaginii se va confunda cu fundalul paginii.

Capitolul 16
Motoare de căutare şi metatag-uri

1. Ce este un motor de căutare?

Dacă aveţi o minimă experienţă în navigarea pe Web v-aţi întâlnit, fără îndoială cu motoarele de căutare. **Motoarele de căutare** sunt aplicaţii plasate pe servere deosebit de puternice, care conţin baze de date imense şi care au drept scop cercetarea şi indexarea sutelor de milioane de pagini Web existente. Utilitatea motoarelor de căutare este evidentă: fără ele ar fi aproape imposibil să găsim informaţiile care ne interesează în oceanul de informaţii care este Web-ul.

Motoarele de căutare conţin un formular de tip câmp de editare în care utilizatorul introduce unul sau mai multe cuvinte folosite drept **cheie de căutare**. Rezultatele sunt furnizate sub forma unei liste de pagini Web care conţin cuvintele cheie introduse. Indexarea unei pagini de către un motor de căutare înseamnă asocierea cuvintelor cheie cu conţinutul informaţional al acesteia. Cele mai folosite motoare de căutare sunt, în acest moment, Google, Yahoo!, Lycos, Go.com şi Excite.

Motoarele de căutare au drept scop indexarea tuturor paginilor existente pe Web, dar, evident, acest lucru este imposibil. Oricât de performante ar fi programele cu care lucrează, numărul de site-uri care sunt publicate în permanenţă pe Web depăşeşte posibilităţile acestora. Acesta este motivul pentru care simpla postare a unui site pe Web nu garantează câtuşi de puţin că acesta va avea vizitatori. Descoperirea lui de către motoarele de căutare poate dura între şase luni şi un an. Tocmai de aceea nu trebuie să aşteptaţi ca site-ul dumneavoastră să fie descoperit, ci trebuie să-l promovaţi activ, una dintre metode fiind înscrierea lui în bazele de date ale motoarelor de căutare.

Există două tipuri principale de motoare de căutare, care se bazează pe moduri diferite de a face indexarea paginilor:

- Motoarele de căutare de tip "păianjen" care realizează indexarea automat, pe baza cuvintelor cheie prezente în antetul documentelor HTML (de exemplu Google). Acest tip de motoare folosesc algoritmi extrem de performanţi, care cercetează şi indexează milioane de pagini pe zi.

- Motoarele de căutare care fac indexarea pe baza cuvintelor cheie furnizate de creatorul paginii, la înscrierea acesteia în baza de date a aplicaţiei (de exemplu Yahoo!). Înscrierea în baza de date se face personal, de către autorul site-ului, iar evaluarea site-ului se face de către personal uman.

Pentru a face mai eficientă indexarea paginilor de către motoarele de tip "păianjen" limbajul HTML include o categorie specială de etichete numite metatag-uri.

2. Metatag-uri

Metatag-urile sunt o categorie specială de etichete care sunt plasate în antetul documentului HTML (în blocul `<HEAD>`). Ca şi celelalte elemente incluse în antetul documentului, cu excepţia titlului documentului, metatag-urile nu sunt vizibile pentru vizitatorul paginii.

Primul metatag despre care vom vorbi este deja cunoscut şi l-aţi folosit de foarte multe ori până acum: este metatagul **`<TITLE>`**. Se utilizează conform sintaxei:

```
<TITLE>Titlul paginii Web</TITLE>.
```

După cum aţi observat din exemplele de până acum, titlul paginii apare în bara de titlu a ferestrei browserului.

Multe dintre motoarele de căutare afişează în lista de rezultate şi titlul paginilor pe lângă adresele URL. Titlul paginii este informaţia care îl determină pe vizitator să intre sau nu pe pagina dumneavoastră. Titlul trebuie să fie o propoziţie care să atragă atenţia şi în acelaşi timp să conţină informaţiile esenţiale privind subiectul şi conţinutul paginii.

Celelalte metatag-uri se introduc prin intermediul etichetei **`<META>`** care nu este o etichetă container. Eticheta `<META>` admite o serie de atribute care oferă motoarelor de căutare diverse informaţii care să ajute la indexarea paginii. Alte atribute oferă informaţii browserului privind data de expirare a documentului sau specifică perioada de timp după care documentul este reîncărcat în mod automat.

Există două categorii de metatag-uri:
1. metatag-urile `name`
2. metatag-urile `http-equiv`

Metatag-urile **`"name"`**

Cele mai folosite metatag-uri sunt cele care se referă la cuvintele cheie care caracterizează pagina şi conţinutul ei, oferind astfel informaţii motoarelor de căutare. Acestea sunt:

```
"keywords"
"description"
"robots"
```

Metatag-ul **`"keywords"`**

Metatag-ul `"keywords"` se introduce conform următoarei sintaxe:

```
<META name="keywords" content="lista cuvinte cheie">
```

Lista cuvintelor cheie trebuie să includă toate cuvintele pe care consideraţi că le-ar putea introduce un vizitator ca cheie de căutare pentru a descoperi pagina dumneavoastră. Cuvintele se scriu despărţite prin virgulă, fără spaţii intre ele. Pe lângă cuvintele cheie, puteţi folosi şi combinaţii de cuvinte. De exemplu într-o pagină care oferă informaţii despre crearea de pagini Web şi web design, cuvintele cheie ar putea fi următoarele:

```
<META name="keywords" content="Web design, creare pagini Web, construire site, Web site design">
```

Găsirea cuvintelor şi combinaţiilor de cuvinte cheie este o etapă care trebuie tratată cu atenţie. Pentru a aduce pagina dumneavoastră în atenţia acelor vizitatori pentru care este ea concepută, cuvintele cheie trebuie alese cu atenţie şi ele trebuie să reflecte exact elementele de conţinut definitorii ale paginii.

Unii dintre creatorii de pagini Web care doresc să obţină o poziţionare mai bună în lista de rezultate furnizate de motoarele de căutare folosesc în mod abuziv cuvintele cheie, repetând acelaşi cuvânt sau combinaţii de cuvinte de un număr foarte mare de ori sau folosind cuvinte cheie care nu au legătură cu subiectul paginii. Ca urmare a acestor încercări de "păcălire" a motoarelor de căutare, au fost dezvoltaţi algoritmi performanţi care identifică tentativele de fraudă şi care se soldează fie cu scăderea cotaţiei paginilor respective, fie chiar cu eliminarea lor din baza de date a motorului de căutare.

Metatag-ul `"description"`

Metatag-ul în cadrul căruia se realizează descrierea paginii este, de asemenea foarte important pentru succesul acesteia. Ca şi cel anterior, se introduce în antetul documentului conform următoarei sintaxe:

```
<META name="description" content="descrierea subiectului paginii">
```

Multe motoare de căutare includ în lista de rezultate pe lângă URL-ul şi titlul paginii şi descrierea ei. Pentru a face descrierea paginii dumneavoastră este bine să folosiţi combinaţii de cuvinte cheie, reunite în fraze atractive şi convingătoare. Fiţi concis în descriere şi plasaţi informaţiile esenţiale la început deoarece este posibil ca descrierile prea lungi să fie trunchiate. Ca şi în cazul alegerii cuvintelor cheie, acordaţi o atenţie deosebită formulării unei descrieri cât mai potrivite şi ţineţi seama că aceasta poate fi singura dumneavoastră şansă de a-l convinge pe utilizator să deschidă pagina pe care aţi creat-o. Descrierea unei pagini poate arăta astfel:

```
<META name="description" content="Tutorial de Web design care va initiaza pas cu pas in procesul de construire al unui site Web. Invatati sa concepeti, sa creati si sa promovati propriul dumneavoastra site.">
```

Pe lângă metatag-ul care face descrierea paginii este recomandată introducerea, la începutul paginii, a unui comentariu în cadrul căruia să inseraţi descrierea. Acest lucru este util deoarece există motoare de căutare care ignoră metatag-urile şi preiau primele cuvinte pe care le găsesc în pagină. Dacă aţi reuşit să formulaţi o descriere sugestivă şi convingătoare, o puteţi folosi chiar ca frază de început în partea vizibilă a paginii.

Metatag-ul `"robots"`

Sunt situaţii când creatorul paginii Web doreşte ca anumite pagini să fie excluse de la indexarea de către motoarele de căutare. De exemplu dacă aţi publicat

site-ul pe Web înainte de a fi terminat (deşi acest lucru este nerecomandat) sau dacă doriţi să reactualizaţi conţinutul anumitor pagini şi doriţi ca până la finalizarea modificărilor paginile să nu fie accesate de motoarele de căutare, puteţi folosi în antetul paginilor respective metatag-ul "robots" astfel:

```
<META name="robots" content="noindex, nofollow">
```

Această construcţie comunică motoarelor de căutare să nu indexeze pagina şi în acelaşi timp să nu urmeze nici una dintre legăturile incluse în ea. Atributul content poate lua următoarele valori:

- all - permite indexarea tuturor paginilor
- none - nu permite indexarea nici unei pagini şi nici urmarea legăturilor din cadrul lor
- index - permite indexarea paginii
- follow - permite urmarea legăturilor din pagină
- noindex - nu permite indexarea paginii
- nofollow - nu permite urmarea legăturilor din cadrul paginii

Alte metatag-uri name utilizate sunt:

  - <META name="author" content="nume_autor">
- Permite specificarea numelui autorului paginii
- <META name="copyright" content="drepturi copyright">
- Permite specificarea drepturilor de autor
- <META name="generator" content="nume_ aplicatie">
  Specifică aplicaţia cu care a fost creată pagina.
  Exemplu:
  <META name="generator" content="Microsoft FrontPage 3.0">

Metatag-urile "http-equiv"

Pe lângă metatag-urile name care oferă informaţii motoarelor de căutare, există metatag-urile http-equiv care oferă informaţii browserului privind documentul pe care urmează să-l deschidă. Iată în continuare o listă a acestor metatag-uri împreună cu funcţiile fiecăruia dintre ele.

Tabel 16. 1

Eticheta	**Funcţia**
<META http-equiv="Content-Type" content="text/html; charset=iso-8859-1">	Acest tip de metatag-uri sunt echivalente cu headerele HTTP. Ele comunică browserului care este tipul paginii şi cum să o afişeze precum şi setul de caractere folosit (opţional)
<META http-equiv="Content-Script-Type" content="text/javascript">	Specifică limbajul de script folosit în document
<META http-equiv="Content-Style-Type" content="text/css">	Specifică limbajul de foi de stiluri folosit în document
<META http-equiv="expires" content="data_ expirarii>	Permite precizarea datei la care documentul poate fi considerat expirat

<META http-equiv="refresh" content="nr_ secunde"> <META http-equiv="refresh" content="nr secunde; url=adresa_noua">	Specifică numărul de secunde după care pagina este reîncărcată automat de către browser. Atributul `content` poate conţine, opţional, şi adresa URL a altei pagini care se va încărca în locul paginii curente. Această opţiune este utilă când doriţi redirectarea vizitatorului către altă pagină. Exemplu: <META http-equiv="refresh" content="5; url=http://www.nouapagina.com">

După cum observaţi, metatag-urile de tip `http-equiv` sunt folosite pentru a transmite browserului diverse informaţii privind documentul HTML. Din această categorie face parte şi metatag-ul care permite specificarea modului de tranziţie între pagini:

```
<META http-equiv="Page-Enter" content="RevealTrans(Duration=nr_secunde,
Transition=cod_tranzitie)">

<META http-equiv="Page-Exit" content="RevealTrans(Duration=nr_secunde,
Transition=cod_tranzitie)">
```

unde: `nr_secunde` reprezintă durata tranziţiei în secunde iar `cod_tranzitie` poate fi unul dintre codurile de mai jos. Fiecare dintre ele realizează tranziţia paginii în alt mod.

Tabel 16. 2

Cod de tranziţie	Cum se face tranziţia
0	Dreptunghi care se închide
1	Dreptunghi care se deschide
2	Cerc care se închide
3	Cerc care se deschide
4	De jos în sus
5	De sus în jos
6	De la stânga la dreapta
7	De la dreapta la stânga
8	Dungi care se deplasează spre dreapta
9	Dungi care se deplasează în jos

10	Dreptunghiuri care se deplasează spre dreapta
11	Dreptunghiuri care se deplasează în jos
12	Pixeli care "dizolvă" pagina
13	Cortină care se închide orizontal
14	Cortină care se deschide orizontal
15	Cortină care se închide vertical

Exemplul 16.1 ilustrează funcţionarea acestui metatag. Puteţi observa modul de tranziţie dintre pagini din Figura 16.1.

Exemplul 16. 1

```
<HTML>
  <HEAD>
    <TITLE>meta1</TITLE>
    <META http-equiv="page-enter" content="RevealTrans(duration=5,
    transition=0">
    <META http-equiv="page-exit" content="RevealTrans(duration=5,
    transition=1">
  </HEAD>
  <BODY bgcolor="red">
    <H1 align="center">Tranzitia intre pagini</H1><HR>
    Pentru a observa modul de tranzitie apasati <A
    href="meta2.html">aici</A>
  </BODY>
</HTML>
```

Puteţi schimba codurile după cum doriţi pentru a observa şi alte moduri de tranziţie.
Documentul *meta2.html* creat în Exemplul 16.2 serveşte doar pentru a exemplifica modul în care se face tranziţia pentru intrarea în pagină.

Exemplul 16. 2

```
<HTML>
  <HEAD>
    <TITLE>meta2</TITLE>
  </HEAD>
  <BODY bgcolor="yellow">
    <H3>Pentru a reveni la pagina anterioara apasati butonul Back al
    browserului</H3>
  </BODY>
</HTML>
```

Atenţie!

Am precizat că metatag-urile sunt incluse în antetul documentului HTML. Atunci când folosiţi cadre în paginile dumneavoastră nu uitaţi să plasaţi metatag-urile în blocul <HEAD> al tuturor documentelor care se deschid în cadre şi nu numai în documentul de definire a cadrelor <FRAMESET>.

3. Rezumat

Motoarele de căutare sunt aplicaţii foarte puternice care au drept scop indexarea şi adăugarea în baza lor de date a sutelor de milioane de pagini Web existente.

Pentru a facilita indexarea paginii de către motoarele de căutare, în antetul documentului HTML se includ nişte elemente speciale numite metatag-uri, prin intermediul etichetei `<META>`.

Cele mai importante metatag-uri sunt cele care furnizează motoarelor de căutare lista de cuvinte cheie asociată paginii şi descrierea acesteia (metatag-urile "keywords" şi "description"), precum şi cel care interzice indexarea paginilor (metatag-ul "robots").

Alte metatag-uri furnizează informaţii browserului privind tipul paginii, intervalul de timp după care pagina este reîncărcată automat, momentul expirării, autorul, programul cu care a fost construită, etc.

Crearea şi publicarea unui site web

Am studiat până acum elementele fundamentale ale limbajului HTML inclusiv modul în care acesta poate fi extins prin folosirea scripturilor CGI, a scripturilor JavaScript şi a foilor de stiluri. Aveţi la îndemână acum instrumentele necesare pentru a crea propriul dumneavoastră site Web. Dar este oare suficient? Răspunsul este nu.

Pentru a crea un site Web de calitate, care să comunice informaţiile în mod logic şi atractiv şi care să atragă vizitatori permanenţi, cunoaşterea limbajului HTML nu este suficientă. Înainte de a începe scrierea codului HTML pentru paginile dumneavoastră trebuie să parcurgeţi alte două etape importante: planificarea site-ului şi stabilirea elementelor de Web design. După ce aţi finalizat documentele HTML care vor face parte din site trebuie să testaţi funcţionarea link-urilor şi aspectul paginilor în diferite browsere. Şi, în fine, după ce toate aceste etape au fost parcurse, urmează publicarea site-ului pe un server Web care să-l facă accesibil tuturor celor care navighează pe Web.

În cele ce urmează vom discuta despre etapele care preced construirea documentelor HTML care vor forma site-ul şi despre cele care urmează după ce codul HTML al paginilor a fost definitivat.

Capitolul 17
Planificarea site-ului

Planificarea este un aspect crucial în procesul de creare a unui site Web, deoarece este etapa în care se iau decizii care vor influenţa designul, implementarea şi, mai târziu, promovarea site-ului. World Wide Web fiind un mediu deschis şi dinamic, planificarea este mai degrabă un proces continuu în care intervin schimbări determinate de înnoirea permanentă a informaţiilor şi de apariţia altora noi.

Atunci când creaţi un site Web trebuie să fiţi conştient că există o serie de factori asupra cărora creatorul site-ului nu are nici un control. Primul pas în procesul de planificare este identificarea acestor factori şi determinarea modului în care aceştia pot afecta structura viitoare a site-ului dumneavoastră.

1. Factori care nu pot fi planificaţi

Factorii pe care creatorul site-ului nu îi poate controla şi prin urmare nu pot face obiectul unei planificări riguroase sunt următorii:

Comportamentul utilizatorului.

Acest factor implică faptul că nu puteţi controla cum va accesa utilizatorul informaţiile din site-ul dumneavoastră. Web-ul este un **sistem permeabil**, aceasta însemnând că un vizitator poate intra în interiorul site-ului nu numai prin pagina de început (pagina home) ci şi printr-o pagină oarecare. Vizitatorul site-ului poate ajunge la el parcurgând o listă de rezultate furnizată de un motor de căutare, sau prin intermediul unui link aflat pe un alt site, sau parcurgând o listă de resurse. Oricare dintre aceste metode îl poate conduce pe vizitator la o altă pagină din interiorul site-ului, şi nu la cea destinată de dumneavoastră a fi pagina Home.

În general, creatorul site-ului este tentat să structureze site-ul având în minte un anumit tip de acces, cel mai adesea cel care pleacă de la pagina de start, ceea ce conduce la alegerea unei anumite structuri, la o anumită organizare a legăturilor între pagini, etc. Trebuie să aveţi permanent în vedere faptul că vizitatorul site-ului poate avea un cu totul alt mod de a parcurge site-ul decât cel gândit de dumneavoastră şi să-i furnizaţi de-a lungul întregului site elemente de navigaţie clare care să-l ajute să se orienteze.

Browserul utilizatorului.

Aşa cum am mai discutat, există o mare varietate de browsere folosite pentru a vizualiza paginile Web. Prin urmare, vizitatorii site-ului vor percepe o imagine diferită a site-ului dumneavoastră, în funcţie de tipul de browser folosit. Creatorul site-ului nu poate şti ce tip de browser foloseşte un anumit vizitator şi deci nu are control asupra modului în care va fi afişat conţinutul paginilor sale. De exemplu, mai există încă utilizatori care folosesc versiuni vechi de Netscape sau Explorer, care nu suportă anumite extensii HTML. Există de asemenea încă în uz browsere text, cum este Lynx, care nu suportă grafica. Dacă veţi plasa informaţii esenţiale în fişiere grafice, de exemplu, aceşti utilizatori nu vor avea acces la ele.

Din acest motiv este bine să stabiliţi de la început care doriţi să fie nivelul de accesibilitate al site-ului dumneavoastră din acest punct de vedere. Un alt aspect este înţelegerea faptului că HTML este un limbaj care este destinat definirii structurii documentului şi nu al modului său de afişare. Este recomandat să evitaţi specificarea în detaliu a aspectului paginilor sau optimizarea paginilor pentru un anumit tip de browser.

Legăturile cu pagini externe.

Într-un site Web există de obicei legături către resurse exterioare site-ului care sunt şi ele în afara controlului dumneavoastră. Paginile referite prin aceste legături îşi pot schimba adresele, făcând astfel ca legătura să nu mai fie valabilă (legături perimate). De asemenea pot exista legături rupte, în cazul când serverul pe care este găzduită pagina respectivă are anumite dificultăţi tehnice. Ţinând seama de imposibilitatea de a controla acest aspect, pot fi adoptate mai multe metode de abordare:

- Site-ul să nu conţină nici o legătură externă sau care să se afle în afara controlului direct al designerului. Este metoda cea mai sigură. Din păcate, această strategie anulează posibilitatea ca vizitatorul să beneficieze de informaţii conexe cu subiectul site-ului dumneavoastră aflate în exteriorul său.

- Centralizarea resurselor. Este o practică des întâlnită pe Web, aceea de a include toate legăturile externe într-o pagină special destinată acestora. Beneficiul acestei strategii este că în cazul când una dintre aceste legături externe nu mai funcţionează, vizitatorul se poate cu uşurinţă reîntoarce la pagina de resurse pentru a testa următorul link. În plus, cu unele excepţii, vizitatorii vor parcurge cel puţin o parte a site-ului înainte de a ajunge la pagina de legături externe şi a-l părăsi, eventual, definitiv.

- Ieşirea liberă. Este cea mai flexibilă abordare, permiţând plasarea legăturilor externe oriunde în cadrul paginilor. Dezavantajul ei major este că vizitatorul poate părăsi prematur site-ul pentru a urma un anumit link.

Dacă nu aveţi control asupra legăturilor de la site către exterior, este tot atât de adevărat că nu puteţi controla nici legăturile care se fac din exterior către site. Acest lucru poate fi dezavantajos, deoarece nu puteţi şti în ce mod este prezentată legătura către site-ul dumneavoastră. Poate că referirea la el este ironică sau răuvoitoare, însă acest lucru este dincolo de controlul dumneavoastră. Tot ceea ce puteţi face este să creaţi un site de calitate şi cu un conţinut valoros, care să convingă prin el însuşi.

Cu toate că elementele prezentate mai sus pot părea descurajante, unele dintre ele prezintă totuşi şi avantaje. Astfel, permeabilitatea Web-ului poate lucra în favoarea designerului, cu condiţia ca acesta să structureze atent informaţiile prezentate şi să ofere suficiente elemente de navigaţie. O pagină sau un grup de pagini din cadrul unui site pot fi folosite ca referinţe în alte pagini sau chiar ca elemente constitutive ale unui alt site. De exemplu, un site de tip mono-pagină care face prezentarea unui manual de HTML poate fi folosit ca pagină individuală într-un site de librărie electronică, sau poate constitui o legătură utilă din interiorul unui site de design Web, mărind astfel şansele ca pagina respectivă să fie accesată de vizitatori printr-unul dintre aceste puncte.

2. Etapele planificării site-ului

Procesul de planificare a unui site trebuie să parcurgă următoarele etape:

- Stabilirea **audienţei**
- Stabilirea **scopului**
- Definirea **obiectivelor**
- Colectarea **informaţiilor** despre subiectul prezentat
- Stabilirea **specificaţiilor**
- Stabilirea **modului de prezentare**

Audienţa

Pentru a crea un site de calitate şi o comunicare eficientă a informaţiilor, o etapă deosebit de importantă este definirea audienţei site-ului. Audienţa reprezintă publicul căruia i se adresează site-ul. Stabilirea audienţei site-ului dumneavoastră este esenţială deoarece contribuie la definirea conţinutului paginilor, precum şi a organizării şi aspectului său. Un site Web creat pe baza unor informaţii precise asupra audienţei sale actuale şi viitoare are mult mai multe şanse de succes decât unul care nu se adresează unui public specific.

Stabilirea audienţei unui site implică doi paşi:

- Definirea publicului ţintă.

  Trebuie să stabiliţi **cui** se adresează site-ul dumneavoastră. Puteţi, de exemplu să vă adresaţi "persoanelor care studiază chimia". Deşi este un enunţ care defineşte într-o oarecare măsură audienţa site-ului, este de dorit ca definirea audienţei să fie mai precisă decât atât. Vă puteţi adresa de pildă, specialiştilor în chimie, şi atunci informaţiile vor avea un înalt grad de specializare şi un nivel ştiinţific ridicat sau vă puteţi adresa elevilor de liceu interesaţi de studiul chimiei, sau care vor da examene la această disciplină. În acest caz informaţiile vor fi de nivel mai general, legate de programa şcolară şi de tipurile de subiecte cu care elevii se vor confrunta la examene.

  Dacă, de exemplu, doriţi să creaţi un site comercial, stabilirea segmentului de piaţă căruia vă adresaţi este la fel de importantă. Dacă intenţionaţi să vindeţi produse cosmetice vă veţi adresa probabil femeilor. Dar aceasta este o reprezentare mult prea generală a publicului dumneavoastră. Aţi putea să fiţi mai specific, stabilindu-vă drept segment de piaţă femeile cu vârsta mai mică de 25 de ani. În această situaţie este mai probabil că vor avea succes produsele de înfrumuseţare mai îndrăzneţe, în vreme ce, dacă vă adresaţi femeilor de vârstă medie, cele mai bine vândute vor fi produsele de întreţinere, cremele anti-rid, etc.

  Concluzia care se impune este că definirea cât mai precisă a audienţei este definitorie pentru toate etapele ulterioare ale elaborării site-ului. Cu cât mai bine definită este audienţa, cu atât şansele de a crea un site de succes sunt mai mari.

- Definirea informaţiilor necesare care privesc publicul ţintă.

  Nu toate informaţiile despre publicul ţintă sunt esenţiale în elaborarea site-ului. Dacă intenţionaţi să vă adresaţi specialiştilor în chimie, care anume caracteristici ale acestora sunt importante pentru dumneavoastră? Nivelul de educaţie? Aria de specializare? Caracteristici personale, precum înălţimea şi greutatea? Evident, exceptând cazul în care intenţionaţi să vindeţi prin intermediul site-ului echipament de laborator, ultimele informaţii sunt inutile. Prin urmare este necesar să identificaţi informaţiile relevante privitoare la audienţa site-ului dumneavoastră.

Scopul

Stabilirea scopului site-ului este etapa în care trebuie să răspundeţi la întrebarea "**de ce?**". De ce doriţi să creaţi acest site? Definiţia scopului site-ului reprezintă tema conducătoare în procesul de construire a acestuia. Un site fără un scop clar şi bine definit lansează un mesaj neconvingător şi ceţos. Vizitatorul se va întreba, fără îndoială, "La ce serveşte acest site?" şi se va grăbi să-l părăsească.

Pentru a defini scopul site-ului trebuie să aveţi în vedere următoarele elemente:

- Aria de cuprindere a subiectului. Să presupunem că doriţi să creaţi un site care să conţină informaţii despre muzica rock a anilor ’60. Într-o asemenea situaţie, nu veţi defini drept subiect al site-ului muzica rock în ansamblu, deoarece o definire atât de vastă depăşeşte cu mult aria de cuprindere a subiectului dumneavoastră.

- Audienţa. În scopul site-ului trebuie să faceţi referire şi la audienţa stabilita anterior. Astfel, scopul site-ului ar putea fi definit în maniera următoare: "Site-ul oferă informaţii iubitorilor muzicii rock a anilor ’60, de vârstă medie".

- Nivelul de detaliere a subiectului. Trebuie să specificaţi dacă vă referiţi doar la grupurile rock reprezentative sau doriţi să faceţi o istorie completă a perioadei.

- Beneficiul sau avantajul vizitatorului. Ce are de câştigat o persoană care vizitează site-ului dumneavoastră? Poate afla informaţii inedite despre formaţiile preferate, sau poate fi informat cu privire la evoluţia lor ulterioară.

Stabilirea scopului site-ului determină deciziile ulterioare ale designerului privind mesajul pe care îl transmite site-ul. Un scop bine articulat serveşte ca jalon pentru toate celelalte etape ale procesului de planificare şi creare a site-ului. Scopul site-ului poate reprezenta chiar prima informaţie care le este oferită vizitatorilor, la intrarea pe prima pagină.

Obiectivele

După ce aţi stabilit audienţa site-ului, care sunt informaţiile privitoare la publicul ţintă, precum şi scopul său, pasul următor constă în combinarea tuturor acestor informaţii şi formularea unor obiective specifice ale site-ului dumneavoastră. Obiectivele reprezintă o detaliere a scopului general al site-ului, conţinând informaţiile specifice care vor conduce la îndeplinirea scopului pentru care a fost creat site-ul. De exemplu, dacă scopul unui site este "să ofere informaţii despre oraşul ZZZ" , acesta poate fi dus la îndeplinire prin intermediul unor obiective specifice şi variate cum ar fi: informaţii despre aşezarea geografică a oraşului, despre dezvoltarea economică, despre viaţa culturală, obiective turistice, etc. Cu alte cuvinte, în vreme ce scopul site-ului comunică ce aveţi de gând să faceţi, obiectivele comunică ce informaţii veţi oferi pentru a vă îndeplini scopul propus.

Spre deosebire de scopul site-ului, obiectivele se pot modifica în timp, pe măsură ce apar noi informaţii despre publicul ţintă sau despre subiectul site-ului, cu ajutorul cărora puteţi susţine mai bine scopul său.

Colectarea informaţiilor despre subiect

Informaţiile referitoare la subiectul site-ului includ atât informaţiile *on-line* cât şi sursele clasice de informaţii. În această etapă veţi colecta nu numai informaţii legate de subiectul site-ului care vor fi prezentate utilizatorului ci şi informaţiile şi cunoştinţele de care aveţi nevoie pentru realizarea site-ului.

Paşii necesari în construirea colecţiei de informaţii necesare sunt:

- Stabilirea informaţiilor necesare, atât cele pe care le veţi furniza vizitatorului cât şi cele care vă sunt necesare dumneavoastră

- Determinarea modului cum veţi obţine aceste informaţii. În această etapă trebuie să identificaţi sursele de documentare. Informaţiile despre subiectul ales le puteţi găsi pe Web, în literatura de specialitate, în diverse baze de date.

- Modul de reactualizare a informaţiilor. Dacă informaţiile pe care doriţi să le prezentaţi sunt dinamice şi se perimează cu repeziciune, trebuie să stabiliţi cum intenţionaţi să le actualizaţi şi care este intervalul de timp între două reactualizări. De exemplu, dacă intenţionaţi să creaţi un site care să prezinte ştiri sau date despre vreme, ele vor trebui reactualizate zilnic sau chiar mai frecvent. Dacă site-ul prezintă informaţii despre istorie, evident că informaţiile

vor fi reactualizate mult mai rar, eventual la apariţia unor noi descoperiri arheologice, de exemplu.

Specificaţiile

Stabilirea specificaţiilor pentru un site reprezintă o detaliere a obiectivelor sale şi definirea unor cerinţe sau a unor restricţii. Specificaţiile descriu în detaliu ce informaţii vor fi oferite în paginile site-ului şi cum vor fi ele prezentate. De exemplu, dacă unul dintre obiectivele unui site este "furnizarea de legături către surse bibliografice referitoare la subiect" , atunci specificaţiile vor preciza care sunt aceste surse bibliografice, care sunt adresele lor URL, câte astfel de adrese vor fi incluse într-o pagină, etc.

Specificaţiile trebuie să identifice toate resursele necesare atingerii obiectivelor: link-uri, fişiere grafice, fişiere de sunet sau video, alte elemente care vor fi incluse în site: formulare, imagini hartă, scripturi. De asemenea, în cadrul specificaţiilor trebuie stabilite şi elementele care nu vor fi incluse în pagini (dacă este cazul). De exemplu, se poate specifica să nu fie folosite anumite extensii HTML, sau formulare, fişiere care să depăşească anumite dimensiuni, etc.

Modul de prezentare

Planificarea modului de prezentare implică o serie de decizii care vor servi drept puncte de reper în etapa de construire efectivă a site-ului. Această etapă poate include:

- Crearea unor *template*-uri pentru site
- Crearea unor mostre de documente HTML, imagini hartă, sau formulare
- Eşalonarea în timp a etapelor de creare a site-ului

3. Rezumat

Prima etapă a procesului de construire a unui site este planificarea. În această etapă trebuie identificaţi factorii care nu pot fi controlaţi printre care se numără: comportamentul utilizatorului, browserul cu care va fi vizualizată pagina şi legăturile cu pagini externe.

Planificarea site-ului parcurge următoarele etape:

- Stabilirea audienţei site-ului
- Stabilirea scopului site-ului
- Definirea obiectivelor site-ului
- Colectarea informaţiilor despre subiectul prezentat
- Stabilirea specificaţiilor site-ului
- Stabilirea modului de prezentare a site-ului

Un site de calitate, care să ofere informaţii valoroase în mod atractiv şi să permită o comunicare eficientă a acestora nu se naşte în mod întâmplător. Fără a avea o privire de ansamblu asupra aspectelor stabilite în faza de planificare, şansele de a realiza un site de bună calitate sunt minime. Dacă veţi începe să scrieţi direct codul HTML fără să treceţi prin etapa de planificare veţi obţine doar o îngrămădire de pagini fără coerenţă, adesea defectuos legate între ele, cu un conţinut neclar şi slab structurat.

Capitolul 18
Designul site-ului

Odată parcursă etapa de planificare, având clare audienţa, scopul, obiectivele şi specificaţiile site-ului, puteţi trece la etapa de creare efectivă. Pentru ca site-ul dumneavoastră să aibă un aspect plăcut, o bună organizare, instrumente de navigare eficiente trebuie să cunoaşteţi şi să aplicaţi regulile fundamentale de Web design.

Când sunteţi în faza de concepere a designului principalul dumneavoastră obiectiv este să creaţi un aspect atractiv şi să oferiţi vizitatorului site-ului un sentiment de satisfacţie, pe măsură ce acesta parcurge paginile. Designul unui site trebuie să echilibreze performanţele browserului, cu estetica şi funcţionalitatea site-ului. În etapa de design sunt luate deciziile de ordin practic care vor conduce la îndeplinirea obiectivelor stabilite: câte imagini sau elemente grafice veţi include în pagină, cât de mult text vor conţine paginile, ce texte sau imagini vor fi folosite drept legături.

1. Principiile designului web

Pentru a lua deciziile corecte în ceea ce priveşte designul unui site trebuie să aveţi în vedere câteva principii de bază:

Asocierea semnificaţiilor. Folosiţi-vă de puterea hypertextului pentru a stabili legături între informaţiile înrudite ca semnificaţie.

Menţinerea competitivităţii. Deoarece Web-ul este un mediu foarte competitiv, asiguraţi-vă că designul site-ului se menţine la cel mai scăzut cost posibil, din punctul de vedere al vizitatorului. Acest cost include timpul de încărcare al paginilor, aplicaţiile suplimentare necesare pentru vizualizarea optimă a paginilor precum şi efortul depus de vizitator pentru a înţelege informaţiile prezentate.

Folosirea eficientă a resurselor. Alegeţi pentru site-ul dumneavoastră acele elemente care vin în întâmpinarea necesităţilor utilizatorului, şi sunt cât mai eficiente posibil din punctul de vedere al dimensiunii fişierelor, al timpului de acces şi al întreţinerii ulterioare.

Concentrarea pe necesităţile utilizatorului. Acesta este, poate, cel mai important principiu de Web design şi, paradoxal, cel mai adesea ignorat. Un site Web nu se construieşte pentru a satisface gustul designerului (sau al clientului pentru care lucrează) şi nici pentru a etala cunoştinţele sale vaste asupra celor mai noi tehnici de programare Web, ci pentru a veni în întâmpinarea nevoii de informaţii a vizitatorilor săi. Focalizarea asupra utilizatorului este prioritatea principală a unui site de calitate.

Înţelegerea permeabilităţii. Acest principiu se referă la înţelegerea şi asumarea faptului că vizitatorul poate accesa un site prin oricare pagină a sa. Din acest motiv este de dorit ca informaţiile din cadrul unei pagini să se auto-susţină fără a depinde de informaţiile din restul site-ului. Dacă acest lucru nu este posibil, este obligatorie prezenţa unor instrumente de navigaţie eficiente care să permită vizitatorului orientarea cu uşurinţă în interiorul site-ului.

Crearea unui aspect plăcut, coerent şi fluent. Paginile site-ului trebuie să ofere impresia unui tot bine organizat, elementele vizuale (icon-uri, elemente de

navigare) trebuie să fie coerente pe tot parcursul site-ului, fiecare pagină trebuie să conţină indicii asupra identităţii site-ului şi asupra scopului ei.

Susţinerea interactivităţii. Chiar dacă nu folosiţi formulare care asigură un grad înalt de interactivitate cu vizitatorii site-ului, este obligatorie prezenţa unor informaţii de contact (adresa de mail a Webmaster-ului, cel puţin) astfel încât utilizatorii să poată obţine informaţii suplimentare sau să poată comunica eventualele probleme apărute la parcurgerea site-ului.

Susţinerea navigaţiei. Asigurarea unor instrumente de navigaţie eficiente este una dintre condiţiile esenţiale ale unui site de calitate. Se spune că un site bun este acela în care vizitatorul nu este niciodată obligat să apese butonul Back al browserului.

2. Organizarea unui site

Buna organizare a site-ului este unul dintre elementele cheie ale succesului său. Modul de organizare depinde de scopul, obiectivele şi subiectul site-ului şi se bazează pe principiile de design enunţate mai sus.

În funcţie de structura lor, site-urile se împart în mai multe categorii:

- site-uri liniare, formate dintr-o singură pagină (mono-pagină)
- site-uri liniare formate din mai multe pagini (multi-pagină)
- site-uri cu structură ierarhică
- site-uri cu structură de tip Web

Legăturile dintre pagini trebuie să fie corespunzătoare tipului de site pe care îl construiţi.

Site-urile liniare mono-pagină

Acest tip de site este, aşa cum indică şi numele, format dintr-o singură pagină. Această structură se foloseşte atunci când informaţiile prezentate sunt sub formă de text care se poate împărţi firesc în secţiuni mai mici. Vizitatorii pot parcurge întreaga pagină derulând-o dar, de obicei, la începutul paginii există o listă de legături care are rol de Cuprins. Acestea sunt nişte legături interne (ancore) care conduc rapid vizitatorul la secţiunea care îl interesează, fără a mai derula întreaga pagină. Împărţirea conţinutului paginii în secţiuni mai mici se poate face folosind linii orizontale. Este indicat ca la fiecare nouă secţiune să inseraţi o legătură internă către partea superioară a paginii unde se află Cuprinsul.

Site-urile liniare multi-pagină

Structura de acest tip se foloseşte în situaţia când informaţiile prezentate se succed într-o ordine secvenţială, de la început la sfârşit, informaţiile prezentate într-o pagină bazându-se pe cele din pagina anterioară. Pentru a-l îndruma pe vizitator să parcurgă site-ul în ordine, fiecare pagină trebuie să conţină o legătură cu pagina următoare, precum şi cu cea anterioară. De asemenea, este necesar să inseraţi şi o legătură cu prima pagină a site-ului care trebuie să conţină Cuprinsul, pentru a facilita şi saltul direct la o anumită pagină. Într-un site cu o astfel de structură paginile nu trebuie să fie prea lungi (de dorit ar fi să nu depăşească un ecran) pentru a face navigarea mai comodă. Cu toate că acest tip de organizare este logică, nu trebuie să uitaţi principiul permeabilităţii. Pentru un vizitator care va intra în site printr-o pagină oarecare, indicaţii de navigare cum ar fi "Înainte" , "Înapoi" ar putea să nu aibă prea mult înţeles.

Site-urile cu structură ierarhică

Site-urile de acest tip sunt cele mai numeroase pe Web. Un astfel de site este format dintr-o pagină de bază (Home) de nivel zero, care conţine legături către alte pagini, fiecare pagină conţinând câte o parte a subiectului site-ului. Fiecare dintre aceste pagini de nivel unu poate avea, la rândul său, legături cu alte pagini, detaliind subiectul şi furnizând informaţii specifice.

Un exemplu de site de acest tip este o librărie virtuală. Pagina Home ar putea conţine legături către diverse categorii: Beletristică, Istorie, Politică, Economie, Calculatoare şi Internet. Dacă un vizitator este interesat de un manual de programare în Perl, el va alege legătura către Calculatoare şi Internet şi va ajunge într-o pagină de nivelul unu unde va găsi legături către Programare, Internet, Hardware. Va alege legătura Programare care va deschide o pagină de nivelul doi care va conţine o listă de titluri, printre care şi manualul căutat.

Atunci când concepeţi structura unui astfel de site, trebuie să acordaţi o mare atenţie organizării logice şi fluente a site-ului. Fiecare pagină trebuie să conţină o legătură către pagina Home astfel încât vizitatorul să poată reveni la început fără să fie obligat să străbată toate nivelele. Nu legaţi prima pagină de prea multe pagini de nivelul unu ci ramificaţi-le în adâncime. Dacă site-ul este de mari dimensiuni, introduceţi în partea superioară a fiecărei pagini o bară de navigare care informează vizitatorul despre locul unde se află. Revenind la exemplul cu librăria, un astfel de instrument de navigare ar putea arăta aşa: *Home > Calculatoare > Programare.*
În plus, este indicat să oferiţi o bară de navigare secundară la baza paginii.

Site-urile de tip reţea.

Aceste site-uri au o structură liberă. Ele sunt formate din mai multe pagini, fiecare putând avea legătură cu oricare altă pagină. Există şi aici o pagină Home, însă de la ea, vizitatorul poate naviga prin site fără a urma un drum precis. În general, acest tip de site este potrivit pentru subiectele care nu au o structură logică internă, subiecte recreaţionale sau distractive. Dacă doriţi să creaţi un site de acest tip, trebuie să aveţi grijă să oferiţi în fiecare pagină, pe lângă legăturile cu alte pagini, o legătură către pagina Home. În plus, asiguraţi-vă că materialul dumneavoastră este adecvat acestui tip de site deoarece altfel site-ul va purta amprenta neconcordanţei între subiectul abordat şi modul său de organizare.

3. Metodologia de construire a site-ului

Deşi nu există un mod unic de desfăşurare a procesului de construire a unui site, există trei tipuri de abordări posibile, pe care creatorul site-ului le poate alege sau combina, în funcţie de necesităţi.

Metoda "Top-Down"

Dacă designerul are încă de la început o idee clară asupra conţinutului site-ului, această abordare este cea mai potrivită. În acest tip de metodologie, este creată mai întâi pagina de început a site-ului (pagina Home) şi apoi celelalte pagini. Paginile pot conţine un minim de informaţii, urmând ca la dezvoltarea ulterioară a site-ului, ele să fie îmbogăţite. Avantajul major al acestei abordări este acela că permite continuitatea vizuală şi de conţinut, deoarece toate paginile vor fi construite în acord cu pagina de start. O metodă foarte bună de a realiza acest lucru este crearea unor *template*-uri care să conţină acelaşi tip de elemente

pentru toate paginile şi care vor fi folosite drept tipare la momentul scrierii codului HTML pentru paginile respective.

Metoda "Bottom-Up"

Această abordare se foloseşte când designerul nu cunoaşte de la început care vor fi structura şi aspectul final al site-ului, dar cunoaşte aspectul şi conţinutul unor pagini din cadrul său. Această situaţie poate apărea când doriţi ca site-ul să conţină pagini deja existente, care au fost create în procesul de dezvoltare al altui site, de exemplu. Chiar dacă nu deţineţi pagini create deja de la care să porniţi, această abordare permite crearea unor pagini individuale care îndeplinesc anumite obiective şi care pot fi legate apoi de o pagină Home. Avantajul abordării "Bottom-Up" este acela că, la construirea paginilor individuale, nu mai sunteţi constrâns la respectarea unui anumit stil, consecvent cu cel din pagina Home. Totuşi, ajustarea ulterioară a paginilor în sensul realizării unui aspect unitar, este necesară.

Metoda incrementării

Această metodă constă în construirea unei pagini de start şi a unor pagini individuale legate de acesta, avându-se în vedere crearea unor pagini intermediare, pe măsura necesităţilor. Metoda se foloseşte atunci când este necesară construirea rapidă a unui site care urmează a fi dezvoltat ulterior, în loc de a-l construi în întregime de la început. Este o metodă nerecomandată începătorilor, deoarece prin adăugirile ulterioare există riscul de a obţine un site defectuos organizat şi lipsit de unitate.

4. Tehnici de design web

Pentru a realiza designul unui site sunt folosite o serie de tehnici cu ajutorul cărora paginile capătă contur, atât din punct de vedere vizual cât şi din punct de vedere al organizării legăturilor şi al amplasării conţinutului în pagini. În continuare sunt prezentate câteva dintre aceste tehnici, fiecare dintre ele referindu-se la câte un aspect al designului site-ului.

Schiţa site-ului

Odată ce v-aţi decis asupra modului în care va fi structurat site-ul este foarte bine ca înainte de a începe scrierea efectivă a codului HTML pentru fiecare pagină în parte, să realizaţi o schiţă a întregului site.

Puteţi folosi în acest scop Microsoft Word sau orice alt procesor de text. Pentru început creaţi un document nou în care listaţi punctele majore pe care doriţi să le acoperiţi în pagina Home.
Faceţi de asemenea o listă a elementelor grafice pe care intenţionaţi să le includeţi, însoţite de indicaţii privind aşezarea lor în pagină. Stabiliţi care sunt paginile de nivel unu şi ce informaţii doriţi să oferiţi în cadrul lor, precum şi paginile subordonate acestora care vor cuprinde detalierea subiectelor anunţate în paginile de nivel unu.

Pentru a avea o imagine cât mai exactă a structurii site-ului încă din faza de schiţare a sa este recomandat să folosiţi bara de instrumente *Outlining* din meniul *Tools* > *Customize* din Word. Aceasta vă permite să stabiliţi nivelul paginilor şi să realizaţi cu uşurinţă ramificarea lor în adâncime.

O altă metodă pentru a realiza schiţa site-ului o constituie graficul (sau harta) site-ului. Dacă nu doriţi să vă complicaţi folosind un editor de text sau un

program de grafică, puteţi realiza o astfel de schiţă şi pe hârtie, cu creionul. Desenaţi câte un dreptunghi pentru fiecare pagină din site, specificând în interiorul său scopul şi obiectivele paginii, elementele şi aranjarea lor în pagină. Apoi uniţi dreptunghiurile prin săgeţi pentru a specifica traseele pe care le poate parcurge utilizatorul. Săgeţile reprezintă de fapt legăturile dintre pagini. Aveţi grijă ca spre pagina Home să indice toate săgeţile, pentru a asigura astfel o legătură cu ea din orice pagină a site-ului. În acest mod puteţi planifica pentru fiecare pagină ce urmează să vadă, să înţeleagă şi să facă vizitatorul acesteia, precum şi unde se poate deplasa din pagina respectivă.

Oricare ar fi metoda folosită pentru a schiţa site-ul, la fiecare pagină trebuie să vă puneţi următoarele întrebări:

- Ce doresc să afle vizitatorul din această pagină?
- Ce doresc să facă vizitatorul în acest moment?
- Ce doresc să simtă vizitatorul parcurgând pagina?
- Unde doresc să meargă vizitatorul în continuare?

Desigur, în ultimă instanţă comportamentul şi impresiile vizitatorului scapă controlului designerului, însă un site în care fiecare pagină dă răspunsuri clare şi limpezi acestor întrebări are foarte multe şanse de a întruni aprecierile pozitive ale vizitatorilor săi.

Pagina de intrare în site (pagina splash)

Există multe site-uri care înainte de pagina Home au o pagină de intrare în site, numită **pagină splash**. Scopul unei astfel de pagini este identificarea rapidă a obiectului site-ului în timp ce se încarcă restul de date. Pagina splash are pentru site acelaşi rol pe care îl are coperta unei cărţi sau prima pagină a unei reviste. Această primă pagină trebuie să se încarce rapid, să aibă un impact vizual puternic şi să comunice esenţialul despre subiectul site-ului sau compania căreia îi aparţine site-ul.

Există opinii divergente în legătură cu folosirea şi utilitatea paginii splash într-un site. Mai ales dacă face apel la elemente multimedia, sunet, grafică, animaţie complexă care încetinesc timpul de încărcare, pagina splash poate aduce mai degrabă deservicii site-ului. În plus, pagina splash poate fi resimţită de vizitator ca o barieră în accesul imediat la informaţiile din interiorul site-ului.

Pagina Home

Deşi aspectul şi conţinutul paginii Home poate varia foarte multe de la un site la altul, există câteva elemente comune prezente în orice pagină de start:

- Identificarea firmei sau companiei căreia îi aparţine site-ul (dacă este cazul)
- Descrierea scopului site-ului
- Descrierea structurii site-ului. Pagina Home are şi funcţia de Cuprins al site-ului oferind indicaţii vizitatorilor asupra subiectelor abordate
- Stabilirea relaţiilor între secţiunile de nivel unu ale site-ului şi cele subordonate lor. Aceasta se realizează prin intermediul barelor de navigare, butoanelor, hărţilor de imagini sau listelor de legături.
- Furnizarea informaţiilor de contact.

Paginile din interior

Paginile de nivel unu reprezintă diviziunile majore ale subiectului general al site-ului. Există tentaţia de a include prea multe informaţii detaliate la acest

nivel. Dacă site-ul acoperă un subiect vast, cu o cantitate mare de informaţii specifice, este bine să lăsaţi detalierea acestora pentru paginile de nivel doi. Paginile de nivel unu trebuie să conţină o descriere succintă a subiectului acoperit precum şi legături către paginile de nivel doi care detaliază fiecare parte a subiectului. O tehnică des utilizată este plasarea resurselor suplimentare în pagini de nivelul trei. De exemplu o pagină de nivelul doi care oferă pe lângă text şi imagini explicative ale unui anumite noţiuni poate fi legată de pagini de nivel trei care conţin imaginile. Când vizitatorul face click pe un anumit text aflat în pagina de nivel doi se deschide pagina de nivel trei cu imaginea explicativă. Avantajul acestei abordări este descongestionarea paginilor de nivelul doi care cuprind detalierea subiectului.

Schiţa dumneavoastră trebuie să cuprindă, pe lângă modul de organizare a paginilor, şi o listă cât mai completă a elementelor pe care doriţi să le includeţi în fiecare pagină (imagini, formulare, fişiere multimedia, etc.).

Fragmentarea informaţiilor

Omul poate procesa o cantitate limitată de informaţii într-o anumită unitate de timp. Din acest motiv, una dintre sarcinile specifice în designul Web este fragmentarea informaţiilor în secvenţe care să nu depăşească posibilităţile de cuprindere ale utilizatorului. Cantitatea de informaţii cuprinsă într-o pagină nu trebuie să-l copleşească pe vizitatorul paginii sau să mărească timpul ei de încărcare. De asemenea, modul de fragmentare al informaţiei trebuie să focalizeze atenţia vizitatorului asupra principalelor subiecte abordate în pagină şi să îl ajute să ia cunoştinţă în mod gradat de subiectul prezentat. Fragmentarea corectă a informaţiilor este în avantajul designerului, deoarece îl ajută să creeze pagini reutilizabile. Dacă fiecare pagină pe care o veţi crea serveşte la îndeplinirea unui anumit scop, puteţi include această pagină fie direct, fie ca o resursă utilă şi în alte site-uri pe care le creaţi ulterior şi care au subiecte conexe cu pagina respectivă.

În cadrul site-ului fragmentarea informaţiilor conduce la stabilirea modului în care va fi detaliat subiectul, la determinarea numărului de pagini necesare pentru acesta precum şi a nivelului paginilor. Deşi divizarea subiectului în părţi mai mici este o tehnică deosebit de utilă, nu abuzaţi totuşi de puterea hypertextului. O fragmentare excesivă a subiectului într-un mare număr de pagini este la fel de obositoare ca şi prezentarea acestuia în bloc. Un design eficient anticipează nevoia vizitatorului de a obţine informaţii suplimentare şi furnizează la momentul oportun legături către paginile care detaliază acel punct.

În cadrul unei pagini modul de fragmentare al informaţiilor ţine de logica internă a subiectului prezentat. Pentru a diferenţia părţile constitutive ale subiectului unei pagini o mare importanţă o are organizarea textului în cadrul paginii.

Cei mai mulţi dintre cei care navighează pe Web obişnuiesc să "scaneze" paginile în căutare de informaţii. Ei citesc mai întâi titlurile, listele, şi primele fraze dintr-un paragraf. Este bine să ţineţi seama de acest lucru când organizaţi textul în pagină.

Conţinutul paginii trebuie să fie cât mai uşor de citit. Folosiţi paragrafe scurte, despărţite prin linii libere, evitaţi frazele prea lungi şi folosiţi în mod judicios titlurile. Accentuaţi părţile pe care vreţi să le scoateţi în evidenţă în text prin îngroşare sau scrierea lor cu altă culoare, dar nu în exces. Abuzul de culori sau de texte scrise cu caractere aldine fac pagina încărcată şi greoaie. Este indicat

să vă limitaţi la un număr redus de tipuri de font. Cele mai apreciate sunt Verdana şi Arial, deoarece sunt lizibile şi elegante. Este mai bine să evitaţi folosirea tipului Times New Roman, deşi este un tip foarte folosit în editarea de texte. Spre deosebire de textele tipărite, pe monitor citirea se face cu 25% mai încet deoarece monitorul adaugă fonturilor un anumit grad de neclaritate (*fuzziness*), mai accentuat la fonturile cu serife, aşa cum este Times New Roman. Nu aranjaţi textul în pagină pe două coloane astfel încât vizitatorul să fie nevoit să revină în partea superioară a paginii pentru a citi ce-a de-a doua coloană. Acest format este potrivit pentru ziare şi reviste dar nu şi pentru o pagină Web deoarece rupe cursivitatea deplasării în cadrul site-ului.

Legarea paginilor

Modul în care veţi realiza legăturile dintre pagini depinde de structura site-ului. Dacă optaţi pentru o structură ierarhică, legăturile din interiorul site-ului vor fi adaptate acestei structuri. Avantajul structurii ierarhice este că oferă utilizatorului posibilitatea unei navigări logice, plecând de la informaţiile generale către cele particulare. Dezavantajul acestei tehnici este că utilizatorul trebuie să urmeze o cale prestabilită pentru a ajunge la o anumită informaţie, care se poate afla câteva link-uri distanţă de pagina Home.

O altă manieră de a realiza legăturile din interiorul site-ului este de a lega fiecare pagină de toate celelalte. Veţi obţine astfel o structură ne-ierarhică care are avantajul că fiecare pagină se află la distanţă de un link de oricare alta, inclusiv de pagina Home. Pentru site-urile de dimensiuni reduse o astfel de structură poate funcţiona bine, însă pentru site-urile cu un mare număr de pagini numărul de legături creşte rapid iar navigarea în interiorul site-ului devine foarte dificilă. În plus, utilizatorul nu are la dispoziţie o cale ierarhică prin care poate ajunge la o informaţie specifică.

Instrumentele de navigare

Pentru a realiza o navigare logică şi eficientă în cadrul site-ului instrumentele de navigare trebuie să fie perfect adaptate modului în care sunt create legăturile dintre pagini şi în plus, să ofere indicii vizuale asupra funcţiei lor. Pentru a crea un aspect vizual unitar al întregului site şi pentru a nu deruta vizitatorul, este recomandat ca instrumentele de navigare să fie aceleaşi în toate paginile.

Stabiliţi de la început aspectul icon-urilor, butoanelor sau textului pe care le veţi folosi drept legături şi păstraţi-l pe parcursul întregului site. Această tehnică oferă site-ului coerenţă şi unitate şi stabileşte imediat identitatea fiecărei pagini în cadrul site-ului. În plus, folosirea repetată a aceloraşi elemente grafice de navigare măreşte eficienţa browserului deoarece acestea sunt încărcate la deschiderea primei pagini şi stocate în memoria *cache*, fără a mai fi necesară încărcarea lor la fiecare nouă pagină.

Instrumentele de navigare trebuie să fie cât mai uşor de înţeles şi localizat. Dacă folosiţi o bară de navigare plasaţi-o acolo unde vizitatorii sunt obişnuiţi să o caute: în partea superioară a paginilor, în stânga sau în dreapta paginii. Folosiţi o bară secundară de navigare la baza paginii, aceasta îl va ajuta pe vizitator să se deplaseze într-o altă secţiune a site-ului fără a mai fi nevoit să deruleze pagina în sus până la bara principală de navigare.

Dacă site-ul este de dimensiuni mari este o idee bună să creaţi o pagină specială care să reprezinte harta site-ului şi să plasaţi legătura către ea în bara principală de navigare din cadrul fiecărei pagini. Aceasta va permite vizitatorului să se orienteze în interiorul site-ului, oricare ar fi pagina prin care a

intrat în site. Harta va conţine legături către toate paginile, grupate conform modului de organizare al site-ului.

De asemenea, dacă folosiţi o structură ierarhică, este utilă prezenţa unui instrument suplimentar care să informeze vizitatorul asupra locului unde se află în cadrul site-ului şi a căii prin care a ajuns aici, în maniera următoare:

Home > Pagina de nivel unu > Pagina de nivel doi > Pagina curentă.

Informaţiile despre site şi pagina curentă

Fiecare pagină a site-ului trebuie să conţină indicaţii privind scopul şi obiectivele site-ului precum şi obiectivele specifice acelei pagini. În faza de planificare aţi colectat informaţii privind publicul ţintă, aţi formulat scopul şi obiectivele întregului site. Acestea reprezintă elementele fundamentale în jurul cărora este construit site-ul. Fiecare pagină a sa are drept scop îndeplinirea unuia dintre obiective. Aceste informaţii trebuie comunicate în pagina respectivă. Vizitatorul nu trebuie forţat să ghicească ce rol are o anumită pagină în cadrul site-ului.

Multe site-uri prezente pe Web conţin o pagină specială cu informaţii de contact precum şi o pagină care conţine informaţii despre site şi creatorii săi (pagina *About*). Totuşi este recomandat ca aceste informaţii să fie incluse şi în subsolul paginilor din interiorul site-ului. Este bine să includeţi aici informaţii despre copyright, data ultimei reactualizări a paginii, despre organizaţia sau compania căreia îi este dedicat site-ul (dacă este cazul) şi informaţii de contact (adresă de mail, număr de telefon, adresă poştală).

Organizarea paginii cu ajutorul tabelelor

Unul dintre aspectele pe care trebuie să le aveţi în vedere la construirea unei pagini este rezoluţia ecranului. Mulţi dintre vizitatorii paginii folosesc rezoluţia de 800x600 de pixeli. O pagină cu lăţimea mai mare de 600 de pixeli poate face necesară derularea pe orizontală a paginii, care pentru mulţi vizitatori este obositoare şi incomodă. Din acest motiv, este recomandat ca tot conţinutul paginii să fie inclus într-un tabel de bază cu lăţimea de 600 de pixeli.

Tabelul de bază

Construirea paginii începe cu un tabel de bază, cu lăţimea (`width`) de 600 de pixeli, o linie şi o coloana. Aliniaţi tabelul la marginea din stânga sau centrat în pagină. Setaţi chenarul (`border`) la valoarea "0" pentru a-l face invizibil pentru vizitator. Întregul conţinut al paginii va fi plasat în interiorul acestui tabel.

Asiguraţi-vă că aţi setat culoarea de fond (`bgcolor`) pentru pagina dumneavoastră. Opţiunea "default" lasă browserul să controleze culoarea fondului. De cele mai multe ori aceasta este alb, dar este mai bine să aveţi certitudinea că ea va apărea la fel în orice browser, aşa încât setaţi `bgcolor="white"` sau orice altă culoare doriţi.

În funcţie de felul cum doriţi să vă organizaţi site-ul şi de scopul acestuia veţi hotărî unde doriţi să plasaţi bara principală de navigare.

Bara de navigare în stânga paginii

Dacă aţi decis să optaţi pentru plasarea barei de navigare în partea stânga (cea mai uzuală aşezare), veţi include în tabelul de bază un tabel cu 2 sau 3 coloane.

Dacă folosiţi un tabel cu 2 coloane setaţi distanţa între celule (`cellpadding`) la valoarea "4" pentru a spaţia textul de marginea tabelului. Puteţi specifica lăţimea celulelor fie în pixeli fie în procente.
Dacă folosiţi un tabel cu 3 coloane, coloana din centru poate fi folosită ca "tampon" , care va separa conţinutul coloanei din stânga de conţinutul celei din dreapta. Setaţi atributul `border` al tabelului la "0" pentru ca marginea să nu fie vizibilă. Odată stabilite aceste atribute, puteţi adăuga în tabel atâtea linii câte sunt necesare sau puteţi include un alt tabel în cel existent.

Bara de navigare în partea superioara

Pentru a construi o bară de navigare în partea de sus a paginii, trebuie să includeţi în tabelul de bază un tabel cu o singura coloana. Setaţi atributele `cellpadding = "4"` şi `border = "0"`. În prima linie a tabelului veţi introduce logo-ul sau titlul site-ului, în linia a doua veţi afişa bara de navigare, iar pe liniile următoare, conţinutul paginii.

Bara de navigare în partea dreapta

Este o abordare mai puţin utilizată. Cel mai adesea este folosită pentru paginile de tip catalog de produse, unde imaginile şi descrierea produselor sunt plasate în coloana din stânga, care ocupă cea mai mare parte a spaţiului.

Folosirea tabelelor pentru organizarea conţinutului paginii este o practică foarte folosită. Tabelele vor păstra imaginile şi textul acolo unde le-aţi plasat. Puteţi include şi alte tabele în interiorul celor pe care le-aţi creat, în funcţie de modul în care sunt organizate informaţiile şi de aspectul pe care doriţi să îl daţi paginii.

5. Greşeli de design

În procesul de creare a unui site pot apărea o serie de greşeli de design, cele mai multe datorate nerespectării tehnicilor şi principiilor prezentate anterior.

Pagini fără legături ("fundături")

Una dintre cele mai frustrante situaţii în care se poate găsi o persoană care navighează pe Web este o pagină fără instrumente de navigare şi fără nici un indiciu privind site-ul căruia îi aparţine. Desigur, site-ul din care face parte pagina poate fi localizat observând prima parte a adresei URL a paginii, care apare în bara de adrese a browserului. Dar a forţa vizitatorul să efectueze această manevră este cu totul nerecomandat. Cel mai probabil acesta nu se va osteni să caute pagina Home a unui asemenea site ci îl va părăsi definitiv. Evitaţi să creaţi asemenea pagini.

Prezenţa unor "fundături" într-un site denotă faptul că designerul nu a înţeles caracterul permeabil al Web-ului şi ideea că un vizitator poate intra în site prin oricare pagină a sa. Pericolul construirii unor astfel de pagini apare mai ales în cazul site-urilor liniare multi-pagină, în care designerul doreşte să conducă vizitatorii de-a lungul site-ului pe un traseu liniar, prestabilit. Aceasta nu înseamnă că acest tip de site-uri nu trebuie folosit, mai ales dacă informaţiile prezentate sunt adecvate acestui mod de prezentare. Trebuie însă acordată atenţie instrumentelor de navigare care trebuie astfel concepute încât vizitatorul să poată ajunge cu uşurinţă cel puţin la pagina Home. În plus, informaţiile despre site şi pagina curentă nu trebuie să lipsească din subsolul fiecărei pagini.

O variaţie a acestor pagini de tip "fundătură" o constituie paginile Home care oferă informaţii prea puţine sau lipsite de semnificaţie. Destul de des pot fi întâlnite pe Web pagini Home care nu oferă nici un fel de indicii asupra structurii sau conţinutului site-ului, fără a mai vorbi de scopul şi obiectivele sale. Instrumentele de navigaţie sunt absente, preferându-se folosirea unor imagini sau texte cu prea puţină semnificaţie pentru a realiza accesul în diferitele secţiuni ale site-ului. Poate că este o abordare avangardistă, dar pentru marea majoritate a vizitatorilor este neplăcut să ajungă într-o asemenea pagină aşa încât este mai bine să respectaţi regulile clasice de design prezentate.

Pagini "uriaşe"

Dacă paginile "fundătură" oferă prea puţine indicii vizitatorului, există şi reversul lor: paginile "uriaşe". Aceste pagini sunt încărcate până la refuz cu informaţii, liste, imagini, link-uri şi alte elemente. Paginile de acest fel ridică două mari probleme:

- Timpul de acces. Mai ales dacă pagina conţine foarte multe imagini, timpul de încărcare al paginii poate fi foarte mare
- Supraîncărcarea cu informaţii. Dacă veţi pune prea multă informaţie într-o singură pagină vizitatorul acesteia nu va fi capabil să o proceseze.

Există opinii care spun că lungimea unei pagini nu ar trebui să depăşească un ecran pentru a se evita ca vizitatorul să deruleze pagina pe verticală. O asemenea cerinţă este prea drastică şi poate fi adesea în dezavantajul prezentării explicite a informaţiilor. Fragmentarea subiectului trebuie să se facă în funcţie de logica lui internă şi nu de reguli arbitrare cum este mărimea ferestrei browserului.

Există situaţii în care paginile lungi îndeplinesc cel mai bine obiectivele urmărite. De exemplu, în paginile de prezentare ale unor produse, pe un site comercial, plasarea descrierii produsului pe două sau mai multe pagini rupe cursivitatea prezentării şi se poate solda cu pierderea unor potenţiali clienţi. Analog, dacă pagina conţine o listă lungă de elemente înrudite, ruperea listei este arbitrară şi dezavantajoasă. Pe de altă parte, aglomerarea de informaţii la nivelul unei singure pagini, mai ales când între secţiunile paginii nu există o legătură intrinsecă face ca pagina să apară ca un amalgam din care vizitatorul va extrage cu greu informaţiile care îl interesează.

Dacă este necesar să folosiţi pagini foarte lungi nu uitaţi să oferiţi pe parcursul paginii ancore care să conducă vizitatorul la începutul secţiunilor principale, precum şi în partea superioara, respectiv inferioară a paginii. Dacă pagina conţine mult text nu o încărcaţi suplimentar şi cu imagini sau cu elemente multimedia deoarece timpul de acces va creşte foarte mult.

Excesul de elemente multimedia

Folosirea fără discernământ a elementelor multimedia (imagine, sunet, video) precum şi folosirea în exces a celor mai noi tehnologii Web este adesea dăunătoare pentru site. Dacă vă adresaţi unei audienţe despre care presupuneţi că nu dispune de conexiuni rapide, de ultimele versiuni de browsere sau de aplicaţiile necesare executării şi afişării corecte a unora dintre elementele din paginile site-ului, este mai bine să nu includeţi aceste elemente.

Astfel, deşi aplicaţiile create cu Macromedia Flash sunt spectaculoase şi se pot realiza efecte speciale deosebit de atractive, un astfel de fişier poate ajunge cu

uşurinţă la dimensiuni care depăşesc 1 megabyte, ceea ce încetineşte considerabil viteza de încărcare a paginii. În plus, pentru vizualizarea corectă a acestora, vizitatorul trebuie să descarce plug-in-ul Shockwave în cazul când nu îl are deja instalat pe computerul propriu. Este foarte neplăcut pentru vizitator să efectueze toate aceste operaţii doar pentru a constata că informaţiile prezentate în această formă puteau fi tot atât de bine prezentate ca text sau imagini obişnuite.
O soluţie de compromis este aceea ca, simultan cu includerea unui astfel de fişier, să se pună la dispoziţia vizitatorului o opţiune de "eludare" de tip "skip intro" , atunci când prezentările sunt folosite doar cu rol de splash screen, ca ecran de deschidere a site-ului, de exemplu.
În plus, folosirea prezentărilor Flash fără ca acestea să servească unui scop bine definit nu face altceva decât să distragă atenţia vizitatorului de la subiectul site-ului.

O altă greşeală destul de des întâlnită este includerea fişierelor audio, fără ca acestea să aibă legătură cu scopul site-ului. Există designeri care consideră că furnizarea unei muzici de fundal sporeşte atractivitatea site-ului. În realitate se întâmplă exact contrariul. În afara situaţiei când subiectul site-ului este legat de muzică sau când fişierele de sunet vin să completeze informaţiile prezentate în pagini, muzica de fundal nu face altceva decât să încetinească încărcarea paginilor şi să-l irite pe vizitator, prin repetare. Dacă este necesar să includeţi fişiere de sunet, lăsaţi vizitatorului posibilitatea de a opta pentru deschiderea lor prin intermediul unei legături, mai ales în cazul în care este vorba despre fişiere de mari dimensiuni.
Este recomandat să procedaţi la fel şi în cazul fişierelor video, avertizându-l pe vizitator că pagina respectivă se va deschide mai greu.

Pagini inegale

Paginile inegale conţin informaţii foarte diferite ca importanţă sau ca nivel de detaliere. Cel mai adesea se confruntă cu această problemă paginile Home în care sunt incluse link-uri către noile pagini create în procesul de dezvoltare a site-ului. Este foarte uşor să cădeţi în greşeala de a include noile legături în pagina Home fără a mai respecta ierarhia stabilită în etapa de organizare a site-ului.

Dacă site-ul la care lucraţi necesită crearea unor noi pagini trebuie să aveţi în vedere plasarea acestora la nivelele specifice obiectivelor pe care le au precum şi încadrarea la locul cuvenit în structura de legături a site-ului prezentată în pagina Home.

Legături fără semnificaţie

Prezenţa legăturilor fără semnificaţie este, de asemenea, o greşeală de design foarte frecventă. Probabil aţi văzut de nenumărate ori în paginile Web formulări ca aceasta:

```
Pentru mai multe informatii apasati <A href="info.html">aici</A>
```

Cuvântul `"aici"` nu are în acest context, nici o semnificaţie. O formulare mult mai potrivită ar putea fi următoarea:

```
Puteti obtine aici <A href="info.html">mai multe informatii</A>
```

O altă situaţie este aceea în care documentul indicat nu are legătură, din punct de vedere al conţinutului, cu pagina care conţine referinţa la el. Toate link-urile

din cadrul unei pagini trebuie să extindă conţinutul informaţional al paginii, vizitatorul care urmează o legătură aşteptându-se ca aceasta să îi ofere informaţii suplimentare despre subiect.

În aceeaşi categorie se pot încadra şi legăturile "banale" , care nu îmbogăţesc prin nimic informaţiile prezentate. De exemplu o construcţie ca aceasta:

```
Bine ati venit in <A href="def.html">Pagina</A> Home a Universitatii
```

unde documentul *def.html* conţine definiţia de dicţionar a cuvântului "pagina" , este o legătură "banală" deoarece în acest context informaţia oferită nu serveşte nici unui scop. Pe de altă parte, într-o pagină al cărei subiect este vocabularul specific Web-ului, o astfel de definiţie ar fi foarte importantă.

Un alt tip de legături fără semnificaţie apare atunci când fragmentarea informaţiei este dusă la extrem prin folosirea unui număr foarte mare de link-uri. Subiectul este secţionat în părţi foarte mici iar vizitatorul este obligat să parcurgă un mare număr de pagini în adâncime pentru a pune cap la cap informaţiile şi a înţelege sensul lor. O abordare de acest tip rupe cursivitatea subiectului şi măreşte efortul de a înţelege şi asimila al vizitatorului.

Pagini dezorganizate şi stridente

Acest tip de pagini suferă din pricina lipsei de organizare a conţinutului. Chiar dacă subiectul este împărţit în secţiuni, unităţile de informaţie sunt împrăştiate în pagină fără o sistematizare prealabilă care să focalizeze atenţia vizitatorului către punctele importante. Aceste pagini au prea multe culori, adesea stridente, tipuri diferite de fonturi, sau imagini de fundal foarte încărcate. Adesea este folosită în exces animaţia fără nici ca aceasta să aibă vreo semnificaţie sau legătură cu conţinutul paginii şi muzica de fundal. Aspectul general al unor astfel de pagini este la fel de strident şi ţipător ca un decor de bâlci.

6. Rezumat

Construirea unui site necesită, pe lângă o bună cunoaştere a limbajului HTML, respectarea unor principii şi folosirea unor tehnici consacrate. Site-ul trebuie să fie orientat către vizitator, către înţelegerea şi satisfacerea nevoii de informaţie a acestuia. Designul trebuie să menţină un echilibru între tehnologiile folosite şi eficienţă, asigurând o viteză convenabilă de încărcare a paginilor şi o vizualizare bună în cele mai populare dintre browsere. Aspectul site-ului trebuie să fie în concordanţă cu scopul şi obiectivele sale, evitând excesele de orice fel.

Principiile care trebuie respectate în etapa de design sunt:
- Asocierea semnificaţiilor
- Menţinerea competitivităţii
- Folosirea eficientă a resurselor
- Concentrarea pe necesităţile utilizatorului
- Înţelegerea permeabilităţii Web-ului
- Crearea unui aspect plăcut, coerent şi fluent
- Susţinerea interactivităţii
- Susţinerea navigaţiei

Strategia de construire a site-ului poate urma una dintre metodele “top-down” , “bottom-up” sau metoda incrementării.

În procesul de design, prima etapă o constituie schiţa site-ului care trebuie să reflecte structura acestuia: liniară, ierarhică sau de tip Web.

Informaţia trebuie fragmentată astfel încât să nu depăşească puterea de asimilare a utilizatorului.

Legăturile dintre pagini trebuie să fie stabilite astfel încât să permită o parcurgere logică a conţinutului site-ului, iar instrumentele de navigare trebuie adaptate modului în care sunt create legăturile dintre pagini, oferind indicii vizuale asupra funcţiei lor. Este recomandat ca instrumentele de navigare să aibă acelaşi aspect în toate paginile.

Nerespectarea principiilor şi tehnicilor de design poate să conducă la construirea unor pagini de tip “fundătura” , a unor pagini prea încărcate cu informaţii sau, dimpotrivă, a unor pagini inegale. De asemenea, printre greşelile de design care pot apărea sunt legăturile fără semnificaţie, excesul de elemente multimedia, paginile dezorganizate sau stridente.

Capitolul 19
Testarea site-ului

Etapa de design a site-ului se desfăşoară în strânsă legătură cu cea de implementare, de scriere a codului HTML pentru paginile care fac parte din site. Totuşi, procesul de creare a site-ului nu se poate considera încheiat în momentul finalizării tuturor documentelor HTML care îl formează. Urmează o etapă adesea tratată oarecum superficial, deşi, ca şi celelalte, este esenţială pentru construirea unui site de calitate: etapa de testare.

1. Corectarea paginilor

Corectarea este unul din cele mai neglijate aspecte al publicisticii Web. Foarte frecvent puteţi întâlni pagini Web cu multiple greşeli de ortografie, gramatică, formatare, chiar şi în cazul site-urilor importante. Paginile cu greşeli de ortografie, exprimări neglijente, reflectă cel puţin o insuficientă experienţă în acest domeniu şi, de ce nu, chiar lipsă de respect pentru vizitator.

Corectarea este neglijată în primul rând datorită uşurinţei extraordinare cu care textele pot fi publicate electronic. Puteţi crea o pagină într-un editor de text şi aceasta poate fi publicată la doar câteva minute după terminarea ei, daca sunteţi suficient de rapid. Cei mai mulţi nu vor petrece ore în şir verificând existenţa eventualelor greşeli gramaticale în textul editat, aşa cum ar face-o, probabil, în cazul unei tipărituri clasice, dacă aceste erori i-ar costa o avere pentru a retipări 10.000 de copii ale documentului. Dacă aţi făcut o greşeală, trebuie doar să deschideţi fişierul, să efectuaţi corectura şi să îl republicaţi pe Web, unde toată lumea îl poate vedea. Este aceasta o strategie corectă? Desigur că nu.

Calitatea muncii pe care aţi depus-o la crearea site-ului defineşte calitatea acestuia. Mii, poate milioane de utilizatori ar fi putut deja citi pagina în cauză. Este mult mai simplu să petreceţi câteva ore corectând textul, pentru a evita ca ulterior să vă luptaţi zile, săptămâni sau luni în şir pentru a vă recâştiga credibilitatea.

Iată câteva metode care vă pot ajuta să corectaţi mai eficient paginile Web:

- Utilizaţi corectoare automate pentru ortografie şi gramatică pentru a descoperi erorile plictisitoare, făcute din grabă sau din oboseală.
- Nu aveţi niciodată încredere absolută în acest tip de corectoare pentru a descoperi erorile mai subtile. După utilizarea lor, încărcaţi pagina în browser şi citiţi-o de câteva ori.
- Pentru site-urile de dimensiuni mari citiţi documentele în mod repetat, căutând de fiecare dată un anumit tip particular de erori
- La prima lectură concentraţi-vă atenţia pe aspectul general al documentului şi pe descoperirea erorilor de formatare, a itemurilor lipsă sau a erorilor de plasare a acestora.
- La a doua lectură verificaţi logica şi cursivitatea ideilor şi a cuvintelor.
- La a treia lectură, corectaţi minuţios întregul text verificând sintaxa, ortografia, punctuaţia.
- Întotdeauna verificaţi imaginile, figurile şi hărţile din pagină. Asiguraţi-vă că ele au legătură cu textul de referinţă, şi verificaţi textul explicativ care însoţeşte imaginea.

Există şi erori ce vor persista uneori chiar după toate aceste verificări. Dacă le descoperiţi după publicarea site-ului, nu mai rămâne decât să le corectaţi atunci.

2. Testarea paginilor

Odată terminată verificarea corectitudinii textului din pagini din punct de vedere gramatical şi logic, urmează etapa de testare a paginilor. În această fază trebuie să vă concentraţi pe testarea corectitudinii etichetelor HTML, a link-urilor, a imaginilor şi a celorlalte elemente incluse în pagini.

Testarea link-urilor

Cea mai simplă cale de testare a link-urilor este de a încărca pagina în browser şi de a da click pe fiecare link.

Verificaţi funcţionarea tuturor ancorelor din pagini care trebuie să acceseze secţiunea din pagină corespunzătoare identificatorului ancorei. Atenţie la secţiunile multiple ale aceleiaşi pagini etichetate cu acelaşi identificator. Această greşeală poate produce rezultate stranii. Daca browserul sare la o altă secţiune a paginii decât cea aşteptată, verificaţi mai întâi identificatorul ancorei în secţiunea pe care browserul o afişează.

Verificaţi apoi modul de funcţionare a legăturilor către alte documente, atât în cadrul site-ului cât şi în afara sa, respectiv validitatea lor şi dacă paginile accesate sunt cele potrivite. Dacă o anumită pagină nu poate fi deschisă, verificaţi sintaxa link-urilor şi anume:

- corectitudinea protocolului specificat
- extensia fişierului
- existenţa simbolului ~

Atenţie!

În UNIX, simbolul tilda (~) este utilizat pentru a specifica directorul de start (*Home directory*) al utilizatorului al cărui nume urmează după acest simbol.

Folosind simbolul ~ vă puteţi referi la o pagină Web astfel:
http://www.server.com/~utilizator/pagina.html

Probleme în funcţionarea paginilor Web

Este posibil ca paginile Web create să nu arate în browser aşa cum au fost ele proiectate. Rezolvarea unor astfel de deficienţe poate fi dificilă, deoarece când scrieţi codul HTML aveţi doar o imagine mentală a modului cum ar trebui să arate pagina. O soluţie pentru a face procesul de depanare mai uşor ar fi să vizualizaţi pagina cu ajutorul browserului pe măsură ce o construiţi. Puteţi izola diversele secţiuni ale paginii pentru a verifica funcţionarea fiecăreia dintre ele.

Cele mai multe greşeli care pot apărea se datorează erorilor în codului sursa al paginii. Sintaxa codului este de o importanta critică în HTML. Cele mai comune probleme de sintaxa care apar sunt:

- absenţa etichetelor de închidere < / >
- împerecherea etichetelor HTML
- ghilimelele " "
- imbricarea etichetelor

Iată câteva dintre problemele cu care vă puteţi confrunta:

O caracteristică de formatare afectează o secvenţă mai mare de text decât era prevăzut.

Să presupunem că aţi folosit într-o secţiune a paginii una dintre etichetele de titlu, <H3> de exemplu. La vizualizarea paginii în browser constataţi că nu doar textul pe care doreaţi să-l evidenţiaţi este formatat astfel, ci o parte mai mare a textului din pagină.
O astfel de situaţie este provocată de:

- Absenţa unuia dintre caracterele " < " sau " > "
- Absenţa etichetei de închidere </H3>
- Imbricarea defectuoasă a etichetelor HTML

O etichetă de închidere căreia îi lipseşte unul dintre caracterele "< "sau ">" nu va fi interpretată corect de browser, prin urmare formatul nu va lua sfârşit decât în punctul în care browserul întâlneşte caracterul respectiv. Aceasta poate determina afişarea într-un format greşit a unui text.

Dacă acest element este un titlu, tot textul dintre eticheta de deschidere asociată titlului şi cea mai apropiată etichetă de închidere întâlnită va fi afişat ca titlu. Daca acest element este text ancoră, tot textul dintre eticheta de deschidere <A> şi prima etichetă de închidere întâlnită va fi afişat ca o legătură (de culoare albastru şi subliniat).

O etichetă de închidere absentă sau o împerechere inadecvata de etichete va provoca probleme similare. Uneori problemele sunt dificil de urmărit şi rezolvat. Urmărirea unei probleme se face în sens descendent, defecţiunea trebuie căutată acolo unde începe secvenţa de cod eronată şi nu unde se sfârşeşte.

O parte din textul paginii nu este afişat

Textul sau obiectele care lipsesc din pagina pot fi depistate prin urmărirea ghilimelelor " " şi a etichetelor greşit închise şi care conţin adrese URL.

Dacă pagina conţine ghilimele de deschidere în interiorul unei etichete, şi cele de închidere trebuie să se regăsească în aceeaşi etichetă. O etichetă din care lipsesc ghilimelele de închidere nu este interpretată corect şi acest lucru poate

produce rezultate bizare. Dacă eticheta căreia îi lipsesc ghilimelele este o etichetă ancoră, întreg textul de la primele ghilimele până la următoarele întâlnite ar putea fi interpretat de browser ca parte dintr-o adresă URL.

O altă eroare care poate provoca acelaşi efect este închiderea greşită a etichetei conţinând o adresă URL. Dacă o astfel de etichetă nu este închisă corespunzător, browserul ar putea interpreta orice text care urmează, până la eticheta de închidere corespunzătoare, ca fiind parte a adresei URL.

Problema se rezolvă prin examinarea etichetelor aflate înaintea porţiunii de text care nu este afişată. Eroarea ar putea proveni de la orice etichetă care conţine ghilimele sau o adresă URL. Ghilimelele utilizate în editorul de text cu ajutorul căruia scrieţi codul HTML trebuie să fie întotdeauna în standard ASCII. Unele editoare de text utilizează aşa-numitele "smart quotes" , în care ghilimele de deschidere arată diferit faţă de cele de închidere. Dacă procesorul de text cu care lucraţi are această posibilitate, ea trebuie dezactivată, deoarece acest tip de ghilimele, nefiind în standardul ASCII, nu vor funcţiona corect în HTML.

Afişarea defectuoasă a formatului

În cazul când formatul afişat de browser nu este cel aşteptat verificaţi mai întâi compatibilitatea dintre browser şi stilul de caractere specificat în codul HTML. Browserul trebuie să fie capabil să afişeze stilul ales. Atunci când sunt folosite stiluri logice, browserul este ultima instanţă în decizia privind stilul caracterelor afişate.

O metodă rapidă de verificare a existenţei unei probleme de compatibilitate este de a afişa paginile utilizând browsere diferite, dintre care unul să poată afişa în mod sigur stilul de caracter utilizat. Dacă acel browser afişează textul incorect, se face verificarea codului HTML. Dacă textul este afişat corect, atunci există o incompatibilitate cu celelalte browsere.
În codul HTML, problema poate fi legată de o imbricare defectuoasă a etichetelor. Verificaţi modul cum au fost imbricate etichetele de formatare din secţiunea de pagină asociată cu eroarea respectivă. Astfel o construcţie de tipul următor:

```
<EM>Va multumim ca ati vizitat <STRONG>Pagina</EM>noastra</STRONG>
```

nu va funcţiona corect, deoarece etichetele <EM> şi <STRONG> sunt imbricate eronat.

3. Rezumat

După etapele de planificare, design şi implementare urmează etapa de testare a site-ului.
În această etapă se realizează:
- Corectarea paginilor din punct de vedere ortografic şi gramatical
- Testarea legăturilor din cadrul site-ului
- Testarea legăturilor externe
- Depanarea greşelilor care ţin de sintaxa codului HTML

Etapa de testare este deosebit de importantă pentru impresia generală lăsată de site-ul dumneavoastră. Un site cu ortografie neglijentă, cu greşeli de exprimare, cu legături nefuncţionale poate îndepărta vizitatorii, oricât de valoroase ar fi informaţiile prezentate în interiorul său.

Capitolul 20
Publicarea site-ului

În sfârşit, site-ul dumneavoastră este finalizat. Aţi parcurs toate etapele, de la planificare, la design şi implementare, v-aţi asigurat că toate elementele funcţionează corect. A venit momentul publicării pe Web a site-ului, astfel încât rezultatele muncii dumneavoastră să fie cunoscute de toţi aceea care vă vor vizita paginile. În etapa publicării pe Web a site-ului intervin mai mulţi paşi:

- Stabilirea unui nume de domeniu
- Stabilirea serverului Web pe care va fi stocat (găzduit) site-ul
- Organizarea şi denumirea fişierelor în conformitate cu cerinţele serverului gazdă
- Transferul fişierelor

1. Numele de domeniu

Pentru ca site-ul dumneavoastră să aibă o identitate pe Web aveţi nevoie de un nume de domeniu pentru el. Numele de domeniu al site-ului va face parte din adresa URL a fiecărei pagini şi va oferi site-ului o prezenţă distinctă pe Web.

Domeniile principale de pe Web pot fi de mai multe tipuri:

- comerciale ***.com***
- educaţionale .***edu***
- guvernamentale .***gov***
- furnizorii de servicii de reţea .***net***
- instituţii non-profit .***org***
- domeniile corespunzătoare ţărilor lumii .***ro***, etc

Pentru a obţine un nume de domeniu pentru site aveţi la dispoziţie două posibilităţi:
- domeniu plătit
- domeniu gratuit

Domeniu plătit

Dacă site-ul pe care l-aţi creat aparţine firmei dumneavoastră sau unei firme client sau dacă doriţi să aveţi o prezenţă stabilă şi credibilă pe Web cea mai bună opţiune este să aveţi un domeniu plătit. Înregistrarea unui domeniu nu este foarte costisitoare şi va asigura site-ului dumneavoastră o identitate serioasă şi credibilă. În plus, odată ce sunteţi proprietarul unui domeniu puteţi să schimbaţi locaţia site-ului (serverul Web pe care este găzduit) fără ca aceasta să necesite schimbarea domeniului.

Preţul pentru înregistrarea unui nume de domeniu variază destul de mult. Una dintre cele mai convenabile oferte poate fi găsită la adresa http://www.10-domains.com care oferă înregistrarea unui nume de domeniu pentru 10 USD anual precum şi o serie de discount-uri pentru înregistrarea pe perioade mai lungi. Dacă doriţi un domeniu românesc (***.ro***) puteţi obţine informaţii la adresa http://www.rnc.ro. Pentru înregistrarea unui astfel de domeniu se percepe o taxă unică de 61 USD (preţ valabil la data scrierii acestei cărţi) fără alte taxe anuale.

În cazul când optaţi pentru un domeniu plătit, adresa URL a site-ului va fi de forma

Domeniu gratuit

Obţinerea unui nume de domeniu gratuit este foarte simplă. Există multe companii *on-line* care oferă astfel de domenii şi, în plus, şi spaţiu de găzduire pentru site. Dacă sunteţi începător şi doriţi să vă testaţi cunoştinţele de Web design nou învăţate, nu este o idee rea să construiţi un site "de probă" folosind pentru acesta un domeniu şi un serviciu de hosting gratuite. În afara acestei situaţii, deşi poate părea alegerea ideală, un domeniu gratuit poate aduce multe dezavantaje site-ului dumneavoastră.

Prima judecată de valoare asupra unui site este făcută pe baza numelui său de domeniu. În cazul unui domeniu gratuit adresa URL a site-ului va fi:

http://www.numefirmagazda.com/~dumneavoastra sau
http://www.dumneavoastra.numefirmagazda.com

Vizitatorii site-ului vor şti imediat că nu deţineţi propriul domeniu şi folosiţi un serviciu de hosting gratuit ceea ce vă va afecta serios credibilitatea. Mai ales dacă site-ul este unul de afaceri, folosirea unui domeniu şi serviciu de hosting gratuit este cu totul contraindicată. În plus, firmele care oferă acest gen de servicii, impun afişarea unor bannere publicitare care distrag atenţia vizitatorilor de la conţinutul site-ului şi măresc timpul de încărcare al paginilor.

2. Serviciul de găzduire Web (Web hosting)

Înainte de a publica site-ul pe Web este necesară o evaluare a necesităţilor de acces, pentru a determina ce fel de cont se potriveşte site-ului dumneavoastră. Plecând de la presupunerea că deţineţi deja o conexiune la Internet, nu vom intra în amănunte privind alegerea unui furnizor de servicii Internet (ISP – *Internet Service Provider*). Totuşi, în cazul când încă nu deţineţi o conexiune sau doriţi să schimbaţi provider-ul actual, vă recomandăm să studiaţi cu atenţie ofertele principalilor furnizori de servicii. Având în vedere că nu vă veţi mai limita doar la simpla navigare pe Web sau la schimbul de emailuri ci veţi dori să transferaţi şi să întreţineţi un site Web necesităţile dumneavoastră vor creşte. Din acest motiv este bine să vă informaţi asupra unor detalii tehnice cum ar fi lăţimea de bandă, mediul de transmisie (antenă satelit, cablu cu fibre optice, unde radio), ce tip de server foloseşte, asistenţa tehnică oferită, dacă oferă găzduire pentru pagini Web.

Opţiunile pentru găzduire sunt următoarele:

- Instalarea unui server Web propriu
- Utilizarea serverului Web al providerului de servicii Internet
- Utilizarea unui server Web aparţinând unei firme care oferă servicii de hosting

Instalarea unui server Web propriu

Instalarea unui server Web propriu este cea mai costisitoare soluţie, dar ea oferă în schimb avantaje semnificative. Dispunând de o conexiune dedicată se pot furniza servicii Web 24 de ore/zi utilizatorilor din lumea întreagă. Veţi dispune de un control complet asupra serverului Web şi puteţi publica orice doriţi. Puteţi configura serverul şi pentru alte servicii, cum ar fi FTP, Gopher, Telnet, scripturi CGI, etc. Puteţi de asemenea avea propriul domeniu care va stabili o prezenţă distinctă pe Web. Adresa URL va avea forma următoare:

http://www.firma_dumneavoastra.com/

Întrucât costurile necesare echipamentului hardware, conexiunii la Internet, configurării şi întreţinerii unui server propriu depăşesc posibilităţile unui utilizator obişnuit, nu vom insista asupra acestei soluţii.

Utilizarea serverului Web al providerului ISP

Utilizarea serverului Web al providerului dumneavoastră de Internet este o opţiune economică. Mulţi dintre furnizorii de servicii Internet oferă în cadrul contului de acces şi un anumit spaţiu pe serverele proprii pe care vă puteţi plasa site-ul, fără a percepe taxe suplimentare. Din păcate, spaţiul oferit este de obicei mic, de ordinul 1-3 Mb şi nu sunt oferite facilităţi pentru crearea de pagini dinamice. Pentru a obţine facilităţi suplimentare va trebui să plătiţi în plus.

Cu toate că fişierele standard, contul de email şi fişierele publicate pe Web utilizează acest spaţiu, 2-3 Mb sunt de regulă suficienţi pentru a menţine un site modest ca dimensiuni. Contul la ISP este accesibil pe baza unei conexiuni dial-up care permite o legătură cu o viteză de până la 56 Kbps.

Înainte de a deschide un astfel de cont, trebuie verificate detaliile privind spaţiul de stocare, taxele pentru spaţiul adiţional, în mod curent 2$ pentru 1 Mb, eventualele alte taxe. Trebuie verificată de asemenea disponibilitatea altor servicii, cum ar fi FTP, Gopher, Telnet, scripturi CGI, care ar trebui să poată fi utilizate gratuit, în cazul în care există.

Un cont la un provider ISP este o opţiune economică dar, în acelaşi timp, limitată. Nu există control al serverului Web, serviciile adiţionale rămânând la latitudinea providerului. Nu veţi avea propriul domeniu, iar adresa URL va arăta astfel:

http://www.provider.com/~dumneavoastra

Utilizarea unui serviciu de hosting

Cea mai bună soluţie din punctul de vedere al raportului servicii/preţ o reprezintă folosirea unui serviciu de hosting plătit. Ofertele de pe piaţa românească sunt numeroase şi variate ca preţ. Înainte de a face o alegere este bine să faceţi un studiu comparativ al acestora în privinţa spaţiului pus la dispoziţie, al modului de transfer al fişierelor, al traficului impus precum şi al prezenţei diverselor facilităţi: scripturi CGI, baze de date, email personalizat, etc. Nu trebuie să pierdeţi din vedere că site-ul dumneavoastră se poate dezvolta astfel încât spaţiul de stocare rezervat la început poate deveni insuficient.

Utilizând un serviciu de hosting adresa URL a site-ului va avea forma următoare:

http://www.dumneavoastra.com

De asemenea veţi beneficia şi de una sau mai multe adrese de email personalizate de tipul:

adresa@dumneavoastra.com

Folosirea serviciilor unei firme de hosting vă permite ca, odată ce deţineţi propriul nume de domeniu, să vă transferaţi site-ul pe serverul Web al firmei şi

să beneficiaţi de o prezenţă stabilă pe Web. Firmele de hosting oferă uneori şi nume de domenii pentru clienţii lor însă există riscul ca dacă vă hotărâţi să renunţaţi la serviciile acelei firme să pierdeţi domeniul. Din acest motiv este mai bine să trataţi separat cele două probleme.

3. Organizarea şi denumirea fişierelor

Deşi această etapă este intrinsecă procesului de construire a site-ului, o vom discuta în acest capitol, deoarece acum este momentul când o bună organizare a directoarelor, subdirectoarelor şi fişierelor devine deosebit de importantă.

Organizarea directoarelor şi fişierelor

Este foarte bine să organizaţi fişierele care fac parte din site pe computerul dumneavoastră exact aşa cum ele vor fi organizate pe serverul Web. Serverul Web are un director "rădăcină" (*root*) unde vor fi stocate toate fişierele site-ului. Folderul unde veţi stoca aceste fişiere pe computerul personal va juca rolul directorului "rădăcină" al serverului. În directorul "rădăcină" va fi plasat fişierul care va conţine pagina "home" a site-ului, fişier denumit de obicei *index.html* sau *index.htm*.

Dacă site-ul este de mici dimensiuni (între 5 şi 10 pagini) puteţi plasa toate fişierele în acelaşi director. Dacă site-ul este mai mare, este recomandat să creaţi pentru fiecare secţiune principală a sa câte un subdirector care va conţine fişierele asociate acelei secţiuni. Subdirectoarele pot conţine la rândul lor câte un fişier index. Atunci când veţi transfera fişierele de pe computerul dumneavoastră pe serverul Web veţi putea să transferaţi întregul subdirector cu fişierele conţinute în el. De asemenea această metodă va face şi întreţinerea site-ului mult mai uşoară.

O altă problemă este locul unde veţi stoca imaginile. Practica standard este de a crea un subdirector special în directorul "rădăcină" unde să plasaţi toate imaginile din site. Dacă procedaţi astfel este extrem de important să creaţi un subdirector similar cu aceeaşi amplasare şi pe computerul dumneavoastră. Calea de la pagina care apelează o imagine la imaginea respectivă trebuie să fie aceeaşi şi pe computerul dumneavoastră şi pe server, altminteri imaginile nu se vor afişa după ce site-ul a fost transferat pe server.

O altă variantă de stocare a imaginilor este crearea unui subdirector destinat imaginilor, plasat în subdirectorul fiecărei pagini. Cu aceleaşi precauţii legate de calea corectă către imagini, această metodă funcţionează bine, ba chiar îmbunătăţeşte viteza de încărcare a imaginilor în pagini. Dezavantajul constă în faptul că adesea este necesar să reţineţi mai multe copii ale aceleiaşi imagini în diferite subdirectoare, în funcţie de paginile unde este folosită.

Verificarea numelor fişierelor

La mutarea fişierelor de pe computerul dumneavoastră pe serverul Web se impune verificarea numelor fişierelor care trebuie să fie compatibile cu sistemul pe sunt mutate.

O atenţie deosebită trebuie acordată denumirii şi extensiei fişierelor. Nu are importanţă dacă optaţi pentru extensia *.html* sau *.htm*. Tot ce trebuie este să fiţi consecvent cu extensia aleasă de-a lungul întregului site. De asemenea, trebuie să aveţi în vedere că în sistemele UNIX, cele mai des folosite ca servere Web, denumirile fişierelor sunt *case-sensitive*. O pagină denumită *mypage.html* nu

este totuna cu pagina *MyPage.html*. Pentru a evita confuziile folosiţi pentru denumirea fişierelor numai litere mici.

4. Transferul fişierelor

Transferul fişierelor care compun site-ul de pe computerul propriu pe serverul Web este o operaţiune relativ simplă, care constă în copierea fişierelor pe server în locaţia destinată site-ului dumneavoastră. Unele dintre firmele de hosting asigură o aplicaţie specială destinată transferului fişierelor, dar cea mai utilizată metodă de transfer este prin intermediul unui client FTP.

Un **client FTP** (*File Transfer Protocol*) este o aplicaţie prin intermediul căreia se poate realiza transferul fişierelor de pe un sistem pe altul. Puteţi copia fişiere de pe computerul personal pe un alt computer (operaţie denumită *Upload*) după cum puteţi prelua fişiere de pe un alt computer, pe computerul personal (operaţie denumită *Download*). Cele mai folosite programe FTP sunt *CuteFTP* (http://www.cuteftp.com) şi *WS_FTP* (http://ipswitch.com). Odată ce aţi instalat pe computerul dumneavoastră un client FTP, v-aţi înregistrat numele de domeniu şi aveţi un cont la un serviciu de găzduire totul este pregătit pentru transferul fişierelor care compun site-ul pe serverul gazdă.

Pentru a realiza transferul fişierelor, în general va trebui să urmaţi următorii paşi:

- Conectarea la Internet
- Deschiderea programului FTP. Veţi observa o serie de casete de dialog în care trebuie introduse informaţiile necesare programului pentru a realiza conexiunea cu computerul gazdă.
- Pentru a face conexiunea cu serverul trebuie să furnizaţi programului adresa FTP a host-ului dumneavoastră. Aceasta vă este furnizată de firma de hosting la deschiderea contului. De asemenea, programul vă va cere numele de utilizator şi parola pe care le-aţi stabilit la deschiderea contului. Nu uitaţi să de-bifaţi opţiunea *Anonymous* din meniul *Login*. Prin introducerea numelui de utilizator şi a parolei veţi căpăta acces la contul dumneavoastră şi veţi putea intra în directorul unde vor fi plasate fişierele site-ului.
- O altă informaţie care trebuie furnizată programului este tipul fişierelor transferate. În general pentru fişierele HTML se foloseşte opţiunea *ASCII* iar pentru celelalte fişiere, opţiunea *binary*. Mai simplu, puteţi alege opţiunea *AutoDetect* prin care programul determină singur tipul fişierului şi modul cum va face transmiterea lui.
- Celelalte casete de dialog pot fi lăsate necompletate sau cu setările preexistente.
- După furnizarea acestor informaţii programul FTP va realiza conexiunea cu serverul gazdă.
- Selectarea fişierelor pe care doriţi să le copiaţi. Veţi observa că fereastra aplicaţiei FTP este împărţită în două: într-o parte aveţi directoarele de pe computerul personal, în cealaltă parte directoarele de pe computerul gazdă. Selectaţi directorul care conţine fişierele dumneavoastră şi daţi comanda de transfer în directorul care vă este destinat pe serverul gazdă.

Dacă folosiţi scripturi CGI asociate site-ului trebuie să le acordaţi o atenţie specială. Cele mai multe servere necesită instalarea scripturilor CGI într-un subdirector special numit *cgi-bin*, aflat în directorul dumneavoastră. Pentru

plasarea şi setarea corectă a scripturilor trebuie să luaţi legătura cu administratorul serverului gazdă care vă va oferi indicaţiile necesare.

Pe unele sisteme de operare modul de accesare al fişierelor este strict definit, permiţând sau restricţionând citirea, scrierea sau executarea fişierelor de către diferiţi utilizatori. Aceste facilităţi sunt setate de regulă prin **drepturile de acces** atribuite fişierelor. Din acest motiv, verificarea modului de setare a fişierelor este extrem de importantă. De exemplu, în sistemele UNIX modul 705 semnifică faptul că fişierele pot fi citite, modificate sau executate de proprietarul lor, în vreme ce alţi utilizatori nu le pot decât citi sau executa (după caz).

După transferul fişierelor, pot apărea probleme de diverse tipuri. Uneori, paginile publicate nu pot fi accesate deloc, scripturile nu funcţionează, etc. Primul lucru care trebuie verificat în acest caz este dacă fişierele se află în directoarele potrivite. Apoi trebuie verificate permisiunile de acces pentru directoare şi fişiere, extensiile fişierelor, fişierele index. În cazul când apar probleme, cea mai sigură cale de rezolvare a lor este să luaţi legătura cu administratorul serverului gazdă şi să îi cereţi informaţii cât mai complete privind drepturile de acces pentru directoare, calea corectă către scripturile folosite şi, în general, setările necesare pentru buna funcţionare a site-ului.

5. Rezumat

În etapa de publicare pe Web a site-ului trebuie parcurşi următorii paşi:
- Stabilirea unui nume de domeniu pentru site: gratuit sau plătit.
- Găsirea unui serviciu de găzduire convenabil sub raportul servicii/preţ.
- Organizarea şi denumirea directoarelor şi fişierelor în conformitate cu cerinţele sistemului serverului gazdă.
- Transferul fişierelor, cea mai folosită metodă de transfer fiind prin FTP.

Câteva cuvinte...la final

Am ajuns, iată, la finalul călătoriei noastre. Aveţi, acum, toate informaţiile şi instrumentele necesare construirii unui site atrăgător, funcţional şi eficient. Puteţi începe prin a vă construi o pagină personală pe care o puteţi plasa pe un server gratuit, pentru a exersa cunoştinţele dobândite. Dar acesta este doar primul pas.

Designul Web este unul dintre domeniile cu un mare potenţial de dezvoltare. Tot mai multe firme doresc site-uri pe Web, fie pentru a derula afaceri prin intermediul lor, fie doar pentru a-şi prezenta oferta de produse şi servicii. O carieră în acest domeniu este o alegere cu mari şanse de succes pentru orice tânăr. Mai mult, deşi pe Web-ul românesc această zonă este încă slab dezvoltată, puteţi să vă iniţiaţi propria afacere care să se deruleze exclusiv pe Internet.

Această carte a urmărit să vă ofere informaţiile necesare unui start corect în domeniul designului Web. Dar nu trebuie să vă opriţi aici. Designul şi programarea Web reprezintă domenii vaste iar cartea noastră a atins doar elementele de bază ale acestora. Puteţi continua învăţând să programaţi în Perl pentru a crea scripturi CGI, puteţi învăţa JavaScript, Java sau un limbaj de interogare pentru bazele de date stocate pe server. Puteţi deveni familiar cu unul din editoarele puternice de HTML (Macromedia Dreamwaver este cel mai utilizat în acest moment) sau puteţi învăţa să stăpâniţi foarte bine un program de grafică (CorelDraw sau Adobe Photoshop de exemplu) pentru a crea grafică Web. Posibilităţile sunt nenumărate.

În lumea World Wide Web totul este într-o continuă mişcare şi dezvoltare. Această carte v-a oferit fundamentele programării în HTML. Continuaţi să învăţaţi, perfecţionaţi-vă continuu şi succesul viitor vă este asigurat.

Anexa 1

Index etichete

Etichetă/ Atribut	Descriere	Browser
`<A></A>`	Creează o legătură (cu atributul `href`) sau identifică o ancoră (cu atributul `name`)	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Href=URL`	Specifică adresa URL a ţintei legăturii	Standard HTML 3.2
`Name=nume`	Specifică identificatorul unei ancore	Standard HTML 3.2
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`Target=nume`	Defineşte numele ferestrei în care se va deschide documentul referit	Explorer, Netscape
`Title=string`	Specifică un titlu pentru fereastra în care se va deschide documentul specificat	Standard HTML 3.2
`<ADDRESS></ ADDRESS>`	Inserează o adresă poştală	Standard HTML 3.2
`<AREA>`	Defineşte o zonă activă într-o imagine hartă	Standard HTML 3.2
`Alt=string`	Specifică un text alternativ pentru browsere*le* non-grafice	Standard HTML 3.2
`Coords=lista coord`	Coordonatele care definesc aria, despărţite prin virgule	Standard HTML 3.2
`Href=URL`	Specifică adresa URL asociată ariei	Standard HTML 3.2
`Shape=forma`	Defineşte forma zonei active	Standard HTML 3.2
`Target=nume`	Specifică numele ferestrei în care se va deschide documentul referit prin aria respectivă	Netscape

`Title=string`	Specifică un titlu pentru aria respectivă	Explorer
`<B></B>`	Textul este scris cu caractere aldine (bold)	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<BASEFONT>`	Precizează caracteristicile fonturilor din documentul curent	Standard HTML 3.2
`Color=culoare`	Specifică culoarea fonturilor	Explorer
`Face=tip fonturi`	Specifică tipul fonturilor	Explorer
`Size=valoare`	Specifică dimensiunea fonturilor	Explorer
`<BIG></BIG>`	Afişează textul cu fonturi mai mari	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<BGSOUND>`	Defineşte fundalul sonor pentru un document	Explorer
`loop=valoare`	Setează numărul de reluări ale bucăţii muzicale. Valori posibile: un număr întreg sau `infinite`	Explorer
`src=URL`	Furnizează adresa URL a fişierului audio.	Explorer
`<BLOCKQUOTE></BLOCKQUOTE>`	Include un citat. Textul este indentat faţă de marginea din stânga a ferestrei	Standard HTML 3.2
`<BODY></BODY>`	Delimitează corpul documentului	Standard HTML 3.2
`Alink=culoare`	Defineşte culoarea legăturii active	Standard HTML 3.2
`Background=URL`	Specifică adresa URL a imaginii de fundal al paginii	Standard HTML 3.2
`Bgcolor=culoare`	Defineşte culoarea fundalului paginii	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape

`Leftmargin=valoare`	Specifică dimensiunea în pixeli a marginii din stânga a paginii	Explorer
`Link=culoare`	Defineşte culoarea legăturilor	Standard HTML 3.2
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`Text=culoare`	Defineşte culoarea textului din pagină	Standard HTML 3.2
`Topmargin=valoare`	Specifică dimensiunea în pixeli a marginii de sus a paginii	Explorer
`Vlink=culoare`	Defineşte culoarea legăturilor vizitate	Standard HTML 3.2
` `	Specifică sfârşitul unei linii şi începutul unei linii noi	Standard HTML 3.2
`clear=margine`	Întrerupe linia curentă şi începe noul rând la marginea dorită. Valori posibile: `left`, `right`, `none`, `all`.	Standard HTML 3.2
`<CAPTION></CAPTION>`	Defineşte titlul unui tabel	Standard HTML 3.2
`Align=pozitie`	Specifică poziţia în care se aliniază titlul. Valori posibile: `top`, `bottom` (Netscape), `left`, `right`, `center` (Explorer)	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`Valign=pozitie`	Specifică poziţia pe verticală a titlului. Valori posibile: `top`, `bottom`	Explorer
`<CENTER></CENTER>`	Centrează textul	Standard HTML 3.2
`<CITE></CITE>`	Inserează un citat	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<CODE></CODE>`	Inserează o mostră de cod	Standard HTML 3.2

`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<COL>`	Defineşte o coloană în cadrul unui grup de coloane	Explorer
`Align=pozitie`	Specifică alinierea coloanei. Valori posibile: `left`, `center`, `right`	Explorer
`Span=n`	Specifică numărul de coloane afectate de eticheta `<COL>`	Explorer
`<COLGROUP></ COLGROUP>`	Defineşte un grup de coloane în cadrul unui tabel	Explorer
`Align=pozitie`	Specifică alinierea pe orizontală a textului din coloane. Valori posibile: `left`, `center`, `right`	Explorer
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer
`Span=n`	Stabileşte numărul de coloane din grup	Explorer
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer
`Valign=pozitie`	Setează alinierea pe verticală a textului din coloane. Valori posibile: `top`, `middle`, `bottom`	Explorer
`Width=valoare`	Setează lăţimea fiecărei coloane din grup, în pixeli sau în procente	Explorer
`<DD></DD>`	Defineşte elementul unei liste de definiţii	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<DFN></DFN>`	Formatează textul inclus ca o definiţie	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<DIR></DIR>`	Creează o listă de directoare, elementele fiind introduse prin eticheta `<LI>`	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape

`Type=marcaj`	Specifică tipul de marcaj. Valori posibile: `circle`, `disc`, `square`.	Explorer, Netscape
`<DIV></DIV>`	Creează o diviziune într-un document	Standard HTML 3.2
`Align=pozitie`	Aliniază elementele incluse în blocul `<DIV>`. Valori posibile: `left`, `center`, `right`	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Nowrap`	Interzice întreruperea liniilor din bloc	Explorer
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<DL></<DL>`	Creează o listă de definiţii	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<DT></DT>`	Introduce un element al unei liste de definiţii	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<EM></EM>`	Accentuează textul inclus între etichete prin scrierea cu caractere italice	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<FONT></FONT>`	Setează caracteristicile textului inclus între etichete	Standard HTML 3.2
`Color=culoare`	Setează culoarea textului	Standard HTML 3.2
`Face=lista tipuri font`	Specifică tipurile de font	Explorer, Netscape
`Size=valoare`	Specifică dimensiunea textului. Valori posibile: 1-7 (valori absolute), +n sau –n (valori relative la dimensiunea curentă)	Standard HTML 3.2

<FORM></FORM>	Defineşte un formular	Standard HTML 3.2
Action=URL	Specifică adresa URL a script-ului care va prelucra datele din formular	Standard HTML 3.2
Class=nume	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
Enctype=mod de codare	Specifică modul de codare al datelor	Standard HTML 3.2
Method=metoda	Specifică metoda de transmitere a datelor. Valori posibile Post, Get	Standard HTML 3.2
Name=nume	Specifică un nume pentru formular pentru a fi folosit de un script JavaScript	Netscape
Style=stil	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
Target=nume	Specifică numele ferestrei în care vor fi afişate rezultatele după procesarea formularului	Explorer, Netscape
<FRAME></ FRAME>	Defineşte un cadru într-un set de cadre	Explorer, Netscape
Border=n	Setează grosimea chenarului unui cadru	Netscape
Bordercolor=culoare	Setează culoarea chenarului unui cadru	Explorer, Netscape
Frameborder=valoare	Stabileşte dacă chenarul cadrului va fi afişat. Valori posibile: yes/no (Netscape), 1 / 0 (Netscape, Explorer)	Explorer, Netscape
Marginheight=n	Setează distanţa în pixeli, dintre conţinutul cadrului şi marginile superioară şi inferioară	Explorer, Netscape
Marginwidth=n	Setează distanţa în pixeli, dintre conţinutul cadrului şi marginile din stânga şi din dreapta	Explorer, Netscape
Name=nume	Specifică numele cadrului	Explorer, Netscape
Noresize	Anulează posibilitatea utilizatorului de a redimensiona cadrul	Explorer, Netscape
Scrolling=valoare	Specifică dacă sunt prezente barele de derulare. Valori posibile: yes, no, auto	Explorer, Netscape
Src=URL	Specifică adresa URL a documentului care se va deschide în cadru	Explorer, Netscape
<FRAMESET></ FRAMESET>	Defineşte un set de cadre	Explorer, Netscape

`Border=n`	Setează grosimea chenarelor cadrelor din set	Netscape
`Bordercolor=culoare`	Setează culoarea chenarelor cadrelor din set	Explorer, Netscape
`Cols=lista`	Specifică numărul şi lăţimea cadrelor din set	Explorer, Netscape
`Frameborder=valoare`	Stabileşte dacă chenarele cadrelor vor fi afişate. Valori posibile: yes/no (Netscape), 1 / 0 (Netscape, Explorer)	Explorer, Netscape
`Framespacing=n`	Defineşte grosimea chenarelor cadrelor din set	Explorer
`Rows=lista`	Specifică numărul şi înălţimea cadrelor din set	Explorer, Netscape
`<Hn></Hn>`	Inserarea titlurilor. *N* ia valori între 1 (titlu de nivel maxim) şi 6 (titlu de nivel minim)	Standard HTML 3.2
`Align=pozitie`	Specifică alinierea titlului. Valori posibile: left, right, center	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<HEAD></HEAD>`	Delimitează antetul documentului	Standard HTML 3.2
`<Hn></Hn>`	Inserează un în text titlu de nivel n, unde n este între 1 şi 6.	Standard HTML 3.2
`Align=type`	Specifică alinierea titlului. Valori posibile: `left`, `center`, `right`.	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<HR>`	Introduce o linie orizontală	Standard HTML 3.2
`Align=pozitie`	Specifică alinierea liniei. Valori posibile: `left`, `center`, `right`	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Color=culoare`	Setează culoarea liniei	Explorer
`Size=valoare`	Specifică grosimea liniei, în pixeli	Standard HTML 3.2

`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`Width=valoare`	Specifică lăţimea liniei, în pixeli sau în procente din lăţimea paginii	Standard HTML 3.2
`<HTML></HTML>`	Delimitează documentul HTML	Standard HTML 3.2
`<I></I>`	Afişează textul cu caractere italice	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<IFRAME></ IFRAME>`	Defineşte un cadru intern	Explorer
`Align=pozitie`	Setează poziţia cadrului faţă de textul care îl înconjoară. Valori posibile: `top`, `center`, `bottom`, `left`, `right`	Explorer
`Frameborder=va loare`	Stabileşte dacă chenarul cadrului va fi afişat. Valori posibile: `1` sau `0`.	Explorer
`Height=n`	Setează înălţimea cadrului, în pixeli	Explorer
`Marginheight=n`	Setează distanţa în pixeli, dintre conţinutul cadrului şi marginile superioară şi inferioară	Explorer
`Marginwidth=n`	Setează distanţa în pixeli, dintre conţinutul cadrului şi marginile din stânga şi din dreapta	Explorer
`Name=nume`	Specifică numele cadrului	Explorer
`Scrolling=valo are`	Specifică dacă sunt prezente barele de derulare. Valori posibile: `yes`, `no`	Explorer
`Src=URL`	Specifică adresa URL a documentului care se va deschide în cadru	Explorer
`Width=n`	Setează lăţimea cadrului, în pixeli	Explorer
`<IMG>`	Inserează o imagine în document	Standard HTML 3.2
`Align=pozitie`	Aliniază imaginea faţă de textul din jur. Valori posibile: `top`, `middle`, `bottom`, `left, right`	Standard HTML 3.2
`Alt=string`	Specifică un text alternativ imaginii	Standard HTML 3.2

`Border=n`	Setează lăţimea în pixeli a chenarului în jurul imaginii	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Controls`	Adaugă butoane de control ferestrei în care rulează un clip video	Explorer
`Dynsrc=URL`	Specifică adresa URL a fişierului video	Explorer
`Height=n`	Specifică înălţimea imaginii, în pixeli	Standard HTML 3.2
`Hspace=n`	Specifică distanţa pe orizontală dintre imagine şi textul din jur	Standard HTML 3.2
`Ismap`	Indică faptul că imaginea este o zonă activă, atunci când este inclusă între etichetele `<A></A>`	Standard HTML 3.2
`Loop=valoare`	Setează numărul de reluări al clipului video. Valoarea poate fi un număr întreg sau `infinite`.	Explorer
`Name=nume`	Furnizează un nume pentru imagine, pentru a fi folosit de un script JavaScript	Netscape
`Src=URL`	Specifică adresa URL a imaginii	Standard HTML 3.2
`Start=valoare`	Specifică momentul începerii clipului. Valori posibile: `FileOpen`, `MouseOver`	Explorer
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`Usemap=URL`	Specifică legătura cu identificatorul hărţii aplicate imaginii	Standard HTML 3.2
`Vspace=n`	Specifică distanţa pe verticală dintre imagine şi textul din jur	Standard HTML 3.2
`Width=n`	Specifică lăţimea imaginii, în pixeli	Standard HTML 3.2
`<INPUT type="button">`	Creează un element de tip buton în cadrul unui formular	Explorer, Netscape
`Name=nume`	Specifică numele parametrului care va fi transmis script-ului care procesează formularul	Explorer, Netscape
`onClick=script`	Specifică script-ul care va fi executat dacă se efectuează click pe buton	Explorer, Netscape

`Value=string`	Specifică textul care va fi afişat pe buton	Explorer, Netscape
`<INPUT type="checkbox">`	Creează un element de tip casetă de validare în cadrul unui formular	Standard HTML 3.2
`Checked`	Marchează elementul ca fiind iniţial selectat	Standard HTML 3.2
`Name=nume`	Specifică numele parametrului care va fi transmis script-ului care procesează formularul, dacă acest element este selectat	Standard HTML 3.2
`onClick=script`	Specifică script-ul care va fi apelat dacă acest element este selectat	Explorer, Netscape
`value=string`	Specifică valoarea parametrului care va fi transmis script-ului care procesează formularul, dacă acest element este selectat	Standard HTML 3.2
`<INPUT type="hidden">`	Creează un element ascuns în cadrul formular	Standard HTML 3.2
`Name=nume`	Specifică numele parametrului care va fi transmis script-ului care procesează formularul	Standard HTML 3.2
`Value=string`	Specifică valoarea acestui element care va fi transmisă script-ului	Standard HTML 3.2
`<INPUT type="image">`	Creează un element de tip imagine în cadrul unui formular	Standard HTML 3.2
`Align=pozitie`	Aliniază elementul faţă de textul care însoţeşte elementele din formular. Valori posibile: `top`, `middle`, `bottom`	Standard HTML 3.2
`Border=n`	Setează grosimea chenarului imaginii, în pixeli	Standard HTML 3.2
`onClick=script`	Specifică script-ul care va fi apelat dacă se efectuează click pe imagine	Explorer, Netscape
`Name=nume`	Specifică numele parametrului care va fi transmis script-ului care procesează formularul	Standard HTML 3.2
`Src=URL`	Specifică adresa URL a imaginii	Standard HTML 3.2
`<INPUT type="password">`	Creează un câmp de tip *password* într-un formular	Standard HTML 3.2
`Maxlength=n`	Specifică numărul maxim de caractere care pot fi introduse	Standard HTML 3.2

Name=nume	Specifică numele parametrului care va fi transmis script-ului care procesează formularul	Standard HTML 3.2
onBlur=script	Specifică un script care va fi apelat când mouse-ul părăseşte elementul	Explorer, Netscape
onChange=script	Specifică un script care va fi apelat când conţinutul câmpului este schimbat	Explorer, Netscape
onFocus=script	Specifică un script care va fi apelat când mouse-ul se află poziţionat pe acest câmp	Explorer, Netscape
onSelect=script	Specifică un script care va fi apelat când se efectuează click pe acest câmp	Explorer, Netscape
Size=n	Specifică numărul de caractere care vor fi afişate în interiorul câmpului	Standard HTML 3.2
Value=string	Specifică valoarea iniţială a câmpului şi valoarea care va fi transmisă script-ului care procesează formularul	Standard HTML 3.2
<INPUT type="radio">	Creează un element de tip buton radio în cadrul unui formular	Standard HTML 3.2
Checked	Marchează elementul ca iniţial selectat	Standard HTML 3.2
Name=nume	Specifică numele parametrului care va fi transmis script-ului care procesează formularul, dacă acest element este selectat	Standard HTML 3.2
onClick=script	Specifică script-ul care va fi apelat dacă acest element este selectat	Explorer, Netscape
Value=string	Specifică valoarea parametrului care va fi transmis script-ului, dacă acest element este selectat	Standard HTML 3.2
<INPUT type="reset">	Creează un buton de tip *Reset* într-un formular	Standard HTML 3.2
onClick=script	Specifică script-ul care va fi executat dacă se efectuează click pe buton	Explorer, Netscape
Value=string	Specifică textul care va fi afişat pe buton (prestabilit acesta este Reset)	Standard HTML 3.2
<INPUT type="submit">	Creează un buton de tip *Submit* într-un formular	Standard HTML 3.2
Name=nume	Specifică numele parametrului care va fi transmis script-ului care procesează formularul	Standard HTML 3.2

`onClick=script`	Specifică script-ul care va fi executat dacă se efectuează click pe buton	Explorer, Netscape
`Value=string`	Specifică textul care va fi afişat pe buton (prestabilit acesta este Submit)	Standard HTML 3.2
`<INPUT type="text">`	Creează un element de tip câmp de editare într-un formular	Standard HTML 3.2
`Maxlength=n`	Specifică numărul maxim de caractere acceptate	Standard HTML 3.2
`Name=nume`	Specifică numele parametrului care va fi transmis script-ului care procesează formularul	Standard HTML 3.2
`onBlur=script`	Specifică un script care va fi apelat când mouse-ul părăseşte elementul	Explorer, Netscape
`onChange=script`	Specifică un script care va fi apelat când conţinutul câmpului este schimbat	Explorer, Netscape
`onFocus=script`	Specifică un script care va fi apelat când mouse-ul se află poziţionat pe acest câmp	Explorer, Netscape
`onSelect=script`	Specifică un script care va fi apelat când se efectuează click pe acest câmp	Explorer, Netscape
`Size=n`	Specifică numărul de caractere care vor fi afişate în interiorul câmpului	Standard HTML 3.2
`Value=string`	Specifică valoarea iniţială pentru acest câmp şi valoarea care va fi transmisă script-ului	Standard HTML 3.2
`<KBD></KBD>`	Textul inclus între etichete este un text de tip "intrare de la tastatură"	Standard HTML 3.2
`<LI></LI>`	Delimitează elementul unei liste	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`Type=format`	Specifică forma de marcaj a elementului.	Standard HTML 3.2
`Value=n`	Setează numărul acestui element în cadrul listei la valoarea *n*	Standard HTML 3.2
`<LINK>`	Defineşte o legătură între documentul în care este prezentă eticheta şi un alt document	Standard HTML 3.2

`Href=URL`	Specifică adresa URL a documentului ţintă	Standard HTML 3.2
`Rel=relatie`	Specifică relaţia dintre documentul curent şi documentul ţintă	Standard HTML 3.2
`Rev=relatie`	Specifică relaţia dintre documentul ţintă şi documentul curent	Standard HTML 3.2
`Title=string`	Specifică un titlu pentru documentul ţintă	Standard HTML 3.2
`Type=string`	Specifică tipul documentului ţintă. Se foloseşte în asociere cu foile de stiluri unde tipul are valoarea text/css	Explorer, Netscape
`<MAP></MAP>`	Defineşte o hartă conţinând zone active în cadrul unei imagini hartă	Standard HTML 3.2
`Name=nume`	Specifică un nume pentru hartă	Standard HTML 3.2
`<MARQUEE></ MARQUEE>`	Creează un text derulant	Explorer
`Align=pozitie`	Aliniază textul derulant fata de textul din jur. Valori posibile: `top`, `bottom`, `middle`	Explorer
`Behavior=tip de miscare`	Defineşte tipul de mişcare. Valori posibile: `scroll`, `slide`, `alternate`	Explorer
`Bgcolor=culoare`	Defineşte un fundal pentru textul derulant	Explorer
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer
`Direction=directia de miscare`	Defineşte direcţia de deplasare. Valori posibile: `left`, `right`	Explorer
`Height=n`	Defineşte înălţimea zonei de text, în pixeli	Explorer
`Hspace=n`	Specifică distanţa pe orizontală între textul derulant şi textul din jur	Explorer
`Loop=valoare`	Setează numărul de animaţii ale textului	Explorer
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer
`Vspace=n`	Specifică distanţa pe verticală între textul derulant şi textul din jur	Explorer
`Width=n`	Defineşte lăţimea ariei de text derulant	Explorer

`<MENU></MENU>`	Defineşte o listă de tip meniu	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`Type=tip marcaj`	Defineşte tipul de marcaj pentru listă. Valori posibile: circle, disc, square	Explorer, Netscape
`<META>`	Furnizează informaţii suplimentare despre document	Standard HTML 3.2
`Charset=set de caractere`	Specifică setul de caractere care va fi utilizat în document	Explorer
`Content=string`	Specifică valoarea meta-informaţiei	Standard HTML 3.2
`Http-equiv=string`	Specifică numele meta-informaţiei, în echivalent HTTP. Provoacă includerea informaţiilor despre document în headerul HTTP care este transmis de server, browserului	Standard HTML 3.2
`Name=nume`	Specifică numele meta-informaţiei	Standard HTML 3.2
`<NOBR></NOBR>`	Interzice întreruperea liniilor în textul inclus între etichete	Explorer, Netscape
`<NOFRAMES></NOFRAMES>`	Defineşte conţinutul furnizat browserelor care nu suportă cadre	Explorer, Netscape
`<NOSCRIPT></NOSCRIPT>`	Defineşte conţinutul furnizat browserelor care nu suportă script-uri	Netscape
`<OL></OL>`	Defineşte o listă ordonată	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Start=n`	Setează numerotarea listei începând de la valoarea *n*	Standard HTML 3.2
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`Type=format`	Setează formatul de marcare a listei. Valori posibile: A (litere mari), a (litere mici), I (cifre romane mari), i (cifre romane mici), 1 (cifre arabe)	Standard HTML 3.2

<OPTION></OPTION>	Defineşte o opţiune în cadrul unui bloc <SELECT>într-un formular	Standard HTML 3.2
Class=nume	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
Selected	Marchează elementul ca fiind iniţial selectat	Standard HTML 3.2
Style=stil	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
Value=string	Specifică valoarea returnată script-ului care prelucrează formularul	Standard HTML 3.2
<P></P>	Specifică începutul şi sfârşitul unui paragraf	Standard HTML 3.2
Align=pozitie	Aliniază textul din paragraf. Valori posibile: left, center, right	Standard HTML 3.2
Class=nume	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
Style=stil	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
<PRE></PRE>	Creează un bloc de text care respectă formatarea predefinită în documentul HTML	Standard HTML 3.2
<S></S> <STRIKE></STRIKE>	Realizează afişarea textului tăiat de o linie orizontală	Standard HTML 3.2
Class=nume	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
Style=stil	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
<SCRIPT></SCRIPT>	Defineşte un script în cadrul unui document	Explorer, Netscape
Language=limba	Specifică limbajul script-ului. Valori posibile: javascript, vbscript	Explorer, Netscape
Src=URL	Specifică adresa URL a documentului care conţine script-ul	Explorer, Netscape
<SELECT></SELECT>	Defineşte un element de tip meniu într-un formular	Standard HTML 3.2
Class=nume	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
Multiple	Permite selectarea de opţiuni multiple din meniu	Standard HTML 3.2

`Name=nume`	Defineşte numele parametrilor corespunzători opţiunilor din meniu. Acestea vor fi transmise script-ului care procesează formularul	Standard HTML 3.2
`onBlur=script`	Specifică un script care va fi apelat când mouse-ul părăseşte elementul	Netscape
`onChange=script`	Specifică un script care va fi apelat când conţinutul câmpului este schimbat	Netscape
`onClick=script`	Specifică un script care va fi apelat când se efectuează click pe acest câmp	Netscape
`onFocus=script`	Specifică un script care va fi apelat când mouse-ul este poziţionat pe acest câmp	Netscape
`Size=n`	Dacă n=1 este afişat un meniu de tip sus/jos, daca n>1 se afişează un meniu cu bară de derulare	Standard HTML 3.2
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<SMALL></SMALL>`	Afişează textul cu fonturi de dimensiune mai mică decât cea curentă	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<SPAN></SPAN>`	Defineşte un sector de text pentru a i se aplica un stil	Explorer, Netscape
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<STRONG></STRONG>`	Afişează textul cu caractere aldine	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<STYLE></STYLE>`	Defineşte un stil în cadrul unui document	Explorer, Netscape
`Type=format`	Specifică formatul stilului. Întotdeauna valoarea este `text/css`	Explorer, Netscape
`<SUB></SUB>`	Permite scrierea indicilor inferiori	Standard HTML 3.2

`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<SUP></SUP>`	Permite scrierea indicilor superiori	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<TABLE></ TABLE>`	Defineşte un tabel	Standard HTML 3.2
`Align=pozitie`	Aliniază tabelul faţă de textul din jur. Valori posibile: left, right, center	Standard HTML 3.2
`Background=URL`	Defineşte o imagine de fundal pentru tabel	Explorer
`Bgcolor=culoare`	Defineşte o culoare de fundal pentru tabel	Explorer, Netscape
`Border=n`	Specifică grosimea în pixeli a chenarului tabelului	Standard HTML 3.2
`Bordercolor=culoare`	Defineşte culoarea chenarului	Explorer
`Bordercolordark=culoare`	Defineşte culoarea laturilor de jos şi dreapta ale tabelului	Explorer
`Bordercolorlight=culoare`	Defineşte culoarea laturilor de sus şi stânga ale tabelului	Explorer
`Cellpadding=n`	Specifică distanţa dintre marginea celulelor şi textul din interiorul lor	Standard HTML 3.2
`Cellspacing=n`	Specifică distanţa dintre celule	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Cols=n`	Specifică numărul de coloane din tabel	Explorer, Netscape
`Frame=pozitie chenar`	Defineşte poziţia laturilor din chenar care vor fi afişate. Valori posibile: `border`, `void`, `above`, `below`, `hsides`, `vsides`, `lhs`, `rhs`, `box`	Explorer
`Height=n`	Defineşte înălţimea tabelului în pixeli sau procente	Standard HTML 3.2

`Hspace=n`	Specifică distanţa pe orizontală între marginile tabelului şi elementele din jur	Netscape
`Nowrap`	Interzice întreruperea liniilor în cadrul celulelor tabelului	Explorer
`Rules=margini`	Specifică unde vor fi afişate chenarele celulelor. Valori posibile: `none`, `rows`, `cols`, `all`	Explorer
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`Valign=pozitie`	Aliniază textul din tabel. Valori posibile: `top`, `center`, `bottom`, `baseline`	Explorer
`Vspace=n`	Specifică distanţa pe verticală între marginile tabelului şi elementele din jur	Netscape
`Width=n`	Defineşte lăţimea tabelului în pixeli sau procente din lăţimea paginii	Standard HTML 3.2
`<TD></TD>`	Defineşte o celulă de date în tabel	Standard HTML 3.2
`Align=pozitie`	Aliniază conţinutul celulei. Valori posibile: `left`, `center`, `right`	Standard HTML 3.2
`Background=URL`	Specifică o imagine de fundal pentru celulă	Explorer
`Bgcolor=culoare`	Specifică o culoare de fundal pentru celulă	Explorer, Netscape
`Bordercolor=culoare`	Defineşte culoarea chenarului celulei	Explorer
`Bordercolordark=culoare`	Defineşte culoarea laturilor de jos şi dreapta ale celulei	Explorer
`Bordercolorlight=culoare`	Defineşte culoarea laturilor de sus şi stânga ale celulei	Explorer
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Colspan=n`	Specifică extinderea celulei curente peste *n* coloane vecine	Standard HTML 3.2
`Height=n`	Defineşte înălţimea celulei în pixeli	Standard HTML 3.2
`Nowrap`	Interzice întreruperea linei de text în celulă	Standard HTML 3.2
`Rowspan=n`	Specifică extinderea celulei curente peste *n* linii vecine	Standard HTML 3.2

`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`Valign=pozitie`	Aliniază conţinutul celulei. Valori posibile: top, center, bottom, baseline	Standard HTML 3.2
`Width=n`	Defineşte lăţimea celulei în pixeli	Standard HTML 3.2
`<TEXTAREA></TEXTAREA>`	Creează un câmp de editare multi-linie în cadrul unui formular	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Cols=n`	Specifică numărul de coloane (caractere) disponibile pe orizontală	Standard HTML 3.2
`Name=nume`	Specifică numele parametrului care va fi transmis script-ului care procesează formularul	Standard HTML 3.2
`onBlur=script`	Specifică un script care va fi apelat când mouse-ul părăseşte elementul	Netscape
`onChange=script`	Specifică un script care va fi apelat când conţinutul câmpului este schimbat	Netscape
`onFocus=script`	Specifică un script care va fi apelat când mouse-ul se află poziţionat pe acest câmp	Netscape
`onSelect=script`	Specifică un script care va fi apelat când se efectuează click pe acest câmp	Netscape
`Rows=n`	Specifică numărul de linii disponibile pe orizontală	Standard HTML 3.2
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`Wrap=tip`	Setează tipul de întrerupere a liniilor de text. Valori posibile: `off`, `virtual` (întreruperea textului e afişată dar nu e transmisă la server), `physical` (întreruperea e afişată şi transmisă la server)	Netscape
`<TH></TH>`	Defineşte un cap de tabel	Standard HTML 3.2
`Align=pozitie`	Aliniază conţinutul celulelor. Valori posibile: left, center, right	Standard HTML 3.2
`Background=URL`	Specifică o imagine de fundal pentru celulă	Explorer
`Bgcolor=culoare`	Specifică o culoare de fundal pentru celulă	Explorer, Netscape

`Bordercolor=culoare`	Defineşte culoarea chenarului celulei	Explorer
`Bordercolordark=culoare`	Defineşte culoarea laturilor de jos şi dreapta ale celulei	Explorer
`Bordercolorlight=culoare`	Defineşte culoarea laturilor de sus şi stânga ale celulei	Explorer
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Colspan=n`	Specifică extinderea celulei curente peste *n* coloane vecine	Standard HTML 3.2
`Height=n`	Defineşte înălţimea celulei în pixeli	Standard HTML 3.2
`Nowrap`	Interzice întreruperea linei de text în celulă	Standard HTML 3.2
`Rowspan=n`	Specifică extinderea celulei curente peste *n* linii vecine	Standard HTML 3.2
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`Valign=pozitie`	Aliniază conţinutul celulei. Valori posibile: top, center, bottom, baseline	Standard HTML 3.2
`Width=n`	Defineşte lăţimea celulei în pixeli	Standard HTML 3.2
`<TITLE></TITLE>`	Defineşte titlul documentului HTML	Standard HTML 3.2
`<TR></TR>`	Defineşte o linie de celule într-un tabel	Standard HTML 3.2
`Align=pozitie`	Aliniază conţinutul celulelor din line. Valori posibile: `left`, `center`, `right`	Standard HTML 3.2
`Bgcolor=culoare`	Specifică o culoare de fundal pentru întreaga linie	Explorer, Netscape
`Bordercolor=culoare`	Defineşte culoarea chenarului liniei	Explorer
`Bordercolordark=culoare`	Defineşte culoarea laturilor de jos şi dreapta ale liniei	Explorer
`Bordercolorlight=culoare`	Defineşte culoarea laturilor de sus şi stânga ale liniei	Explorer
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Nowrap`	Interzice întreruperea liniilor de text pentru toate celulele liniei	Explorer

`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`Valign=pozitie`	Aliniază conţinutul celulelor din cadrul liniei. Valori posibile: `top`, `center`, `bottom`, `baseline`	Standard HTML 3.2
`<TT></TT>`	Formatează textul inclus între etichete în stil monospaţiat	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<U></U>`	Afişează textul subliniat	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`<UL></UL>`	Defineşte o listă neordonată	Standard HTML 3.2
`Class=nume`	Specifică o clasă de stiluri care controlează aspectul acestei etichete	Explorer, Netscape
`Style=stil`	Specifică un stil *in-line* pentru etichetă	Explorer, Netscape
`Type=format`	Defineşte formatul marcajului pentru elementele listei. Valori posibile: `circle`, `disc`, `square`	Standard HTML 3.2

Anexa 2
Dicţionar de termeni

applet (~ Java)	Un program Java de dimensiuni reduse care poate fi inclus într-un document HTML. Applet-urile diferă de programele Java prin aceea că sunt restricţionate din punct de vedere al accesului la anumite resurse ale computerului pe care rulează, precum şi al accesului la resursele reţelei. Aceste restricţii sunt impuse pentru a preveni încălcările accidentale sau deliberate ale securităţii sistemului.

ASCII	Sistemul standard de codificare a caracterelor folosit pentru a reprezenta literele mari şi mici, cifrele, caracterele speciale de pe tastatură, etc. Sistemul ASCII cuprinde 128 de coduri, fiecare dintre acestea fiind reprezentat printr-un număr binar de 7 cifre, între 0000000 şi 1111111. Sistemul ASCII extins cuprinde 256 de coduri, ultimele 128 fiind folosite pentru a reprezenta caractere nestandard: litere accentuate sau care nu fac parte din alfabetul latin, de exemplu.
bps (bits per second)	Unitatea de măsură a vitezei de transfer a datelor. Un modem de 56K poate transmite datele cu o viteză de până la 57 000 bps.
browser	Un program care permite vizualizarea hypertextului şi navigarea între diferite resurse aflate pe Web.
CGI (Common Gateway Interface)	Un set de reguli care descrie modul de comunicare dintre un server Web şi anumite aplicaţii aflate pe acesta, numite programe CGI. Orice aplicaţie poate fi un program CGI dacă respectă regulile de intrare a datelor, respectiv de afişare a rezultatelor impuse de standardul CGI.
cgi-bin	Numele atribuit directorului de pe serverul Web în care sunt stocate programele CGI.
client (computer ~ aplicaţie ~)	Un computer sau o aplicaţie software care este folosit pentru a contacta şi a obţine anumite date de la un alt computer sau aplicaţie (server).
dial-up	O conexiune temporară între două computere prin intermediul liniei telefonice.
director	Un nod în sistemul ierarhic de fişiere, care poate conţine alte noduri: fişiere sau alte directoare.
download	Procesul de transfer al unui fişier de pe un alt computer pe computerul propriu.
domeniu (~ Internet)	Un grup de computere din reţeaua Internet care împart acelaşi nume de domeniu. Numele de domeniu sunt formate din două sau mai multe părţi separate prin punct. Partea din stânga este cea mai specifică, cea din dreapta este cea mai generală. Un computer poate face parte din mai multe domenii (poate avea mai multe nume de domeniu), dar un nume de domeniu indică în mod unic spre un singur computer.

e-mail (electronic mail)	Un sistem de transmitere a datelor (mesaje tip text, imagini, HTML, etc.) între doi sau mai mulţi utilizatori, pe reţeaua Internet. Poştă electronică.
folder	Sinonim, în sistemele Windows, cu director.
FTP (File Transfer Protocol)	Metoda de transfer a fişierelor între două locaţii de pe Internet. Prin FTP un computer poate fi conectat la un alt computer de pe Internet pentru a se realiza trimiterea sau primirea fişierelor. Sistemul FTP a fost inventat şi utilizat cu mult înainte de World Wide Web.
GIF (Graphic Interchange Format)	Unul dintre cele mai folosite formate pentru codificarea imaginilor, folosit mai ales în cazul imaginilor cu număr redus de culori (256) şi care conţin zone extinse colorate cu aceeaşi culoare.
Gopher	Un sistem ierarhizat de meniuri care conţin informaţii despre resursele disponibile pe Internet. Gopher foloseşte o interfaţă de tip text şi funcţionează pe principiul client/server, ceea ce înseamnă că pentru a putea accesa sistemul este necesară o aplicaţie Gopher client. Deşi a cunoscut o largă popularitate la mijlocul anilor '90 în prezent este în general înlocuit de World Wide Web. Astăzi nu mai există decât relativ puţine servere Gopher.
header	Partea de la începutul unui pachet de date care conţine informaţii despre conţinutul pachetului, adresa sursei şi a destinaţiei, verificări ale erorilor şi alte câmpuri.
host	Un computer dintr-o reţea care este furnizorul unor servicii pentru celelalte computere din reţea cum ar fi SMTP (e-mail) sau HTTP (WWW)
HTML (Hypertext Markup Language)	Limbajul folosit pentru a crea documente Web. Documentele HTML pot fi vizualizate folosind un browser.
HTTP (Hypertext Transfer Protocol)	Protocolul de transfer al fişierelor de tip hypertext. Este cel mai important protocol folosit în World Wide Web.
hypertext, hypermedia	Un text care conţine legături (link-uri) către alte documente. Extinderea hypertextului prin includerea elementelor media: sunet, imagini, video este cunoscută sub numele de hypermedia.

Internet	O imensă reţea de reţele de computere interconectate, care acoperă întreg globul.
IP (Internet Protocol)	Un număr format din patru grupuri de numere mai mici de 256, separate prin punct care identifică în mod unic fiecare computer conectat la Internet.
ISP (Internet Service Provider)	O instituţie care furnizează contra cost acces la reţeaua Internet.
Java	Un limbaj de programare orientat pe obiecte, inventat de firma Sun Microsystems, folosit adesea pentru a realiza aplicaţii pentru Internet. Limbajul Java este utilizat foarte frecvent pentru a crea aplicaţii de dimensiuni reduse numite applet-uri care pot fi incluse în paginile Web.
JavaScript	Un limbaj de programare utilizat pentru face paginile Web dinamice şi interactive.
JPEG (Joint Photographic Experts Group)	Un format foarte utilizat pentru fişierele grafice. Este folosit pentru imagini fotografice cu număr foarte mare de culori.
legătura (link)	O referinţă aflată într-un document către un alt punct din acelaşi document sau către un document diferit. La efectuarea de click cu mouse-ul pe legătură, browserul afişează documentul specificat în referinţă.
MIME (Multipurpose Internet Mail Extensions)	La origine, standardul care defineşte tipurile de fişiere ataşate unui mesaj prin e-mail. Este folosit pe scară largă pentru a defini diversele tipuri de fişiere transmise prin reţea.
motor de căutare (search engine)	Un sistem de aplicaţii folosit pentru a căuta informaţii în World Wide Web. Exemple: Google, Yahoo, Lycos, etc.
port	1. Locul de intrare sau ieşire a informaţiilor în/din computer. De exemplu, un port serial este locul unde este conectat modemul. 2. Un număr care face parte uneori din adresa URL, semnificând un anumit serviciu oferit de server. Fiecare serviciu oferit de server deţine un număr standard de port, de exemplu HTTP deţine portul 80.

protocol (~ Internet)	Un set de reguli formale care reglează transferul de informaţii prin reţeaua Internet. Protocoalele de nivel scăzut definesc standardele de tip electric sau fizic iar protocoalele de nivel înalt definesc standardele privind formatarea datelor, sintaxa mesajelor, setul de caractere, etc.
plug-in	O aplicaţie de dimensiuni reduse care adaugă anumite caracteristici sau extinde într-o anumită direcţie posibilităţile unei aplicaţii mai largi.
RGB (Red, Green, Blue)	Sistemul standard de codificare a culorilor. Fiecare culoare este desemnată printr-o combinaţie de trei numere scrise fie în sistemul zecimal fie în cel hexazecimal, care specifică ce cantitate de roşu, verde şi albastru intră în componenţa culorii respective.
script	Un program scris într-un limbaj de programare cum ar fi C/C++, Perl, JavaScript şi altele, care poate fi executat de browser (client-side script) sau de server (server-side script)
server	1. Un computer care furnizează anumite servicii altor computere conectate în reţea. 2. O aplicaţie care asigură anumite servicii altor aplicaţii numite clienţi. Legătura dintre client şi server are drept scop transmiterea de informaţii respectând un anumit protocol.
SMTP (Simple Mail Transfer Protocol)	Protocolul principal utilizat pentru a transmite informaţii de tip e-mail de la un server la altul
TCP/IP (Transmission Control Protocol/Internet Protocol)	O suită de protocoale care definesc în ansamblu transmiterea datelor pe Internet. La origine, a fost destinat sistemului de operare UNIX, în prezent este inclus în toate sistemele de operare importante.
UNIX	Cel mai utilizat sistem de operare folosit pentru serverele de Internet.
upload	Procesul de transfer al datelor de pe computerul personal pe un alt computer.
URL (Uniform Resource Locator)	Adresa unei resurse disponibile pe Internet. Prima parte a adresei se numeşte schemă. Schema precizează protocolul de transfer al datelor, cea utilizată în cadrul Web fiind http.

World Wide Web (WWW)	O colecţie imensă de documente de tip hypertext conectate prin intermediul reţelei Internet şi care foloseşte protocolul HTTP pentru transferul informaţiilor. Prescurtări uzuale: Web, WWW, W3.